Social Media Sentiment Analysis with Explainable AI

No Author Given

No Institute Given

Abstract. Human brain can detect the underlying sentiment of a text or speech without much difficulty. Up until recent times, sentiment analysis was a challenging task for machines. Sentiment analyser are useful tools to detect the sentiments of a text body such as reviews, opinions, blogs, recommendations, ratings, etc. With the advancement of deep learning, sentiment analysis has now become more or less feasible for machines without human intervention. In this paper, we have used a pre-trained model and Bidirectional Long Short Term Memory (Bi-LSTM) Recurrent Neural Network architecture, capable of learning long-term dependencies, to successfully classify text sentiments. We have utilized a dataset containing to 1.6 million tweets retrieved from the social media platform to train and validate this model. After which the model is able to successfully annotate positive and negative sentiment and depressive texts. We also have implemented Explainable AI (XAI) to get a deeper insight into the reason behind our model's decisions that it produces.

Keywords: Bi-LSTM · Word2Vec · Sentiment Analysis · Word Embeddings · Explainable AI · Machine Learning.

1 Introduction

Individuals associate amongst themselves by a multitude of languages where every language is a medium through which they tend to convey and express their thoughts. And whatever they say is accompanied by a feeling, which can be denoted as a sentiment. The term 'sentiment analysis' refers to the process of determining a person's point of view or mood in a given scenario. It simply implies interpreting and determining the thoughts or perception behind a text, voice, or any other forms of interaction [1]. By doing sentiment analysis, we detect, retrieve, or classify the qualitative message conveyed in a phrase using natural language processing.

Social media is accessible to all global virtual platforms where people may express and share their personal thoughts regarding something that has made an impact on their own personal lives. The emergence of online social networks, discussion forums, tweets, blogs has resulted in an enormous increase of data available for sentiment analysis. Posts, articles, tweets, check-ins, and other data reflects the viewpoints on a spectrum of subjects and situations and provide a wealth of material to research and evaluate human sentiment [2].

Today, people have become more outspoken than ever in history because of social media, and on top of that, because of being accessible to all. People confide their true feelings on social media because it serves as a virtual public platform, even if they cannot do so in person.

Besides sharing content, people express their thoughts, ideas, and opinions about a range of events, concepts, and circumstances on platforms like Twitter, Facebook, Reddit, Instagram, etc.

We have used a dataset from Kaggle that had 1.6 million tweets [3]. We have utilized the entire dataset. Texts from the dataset is pre-processed to clean up the tweets and replace tokens with meaningful information for the model. We have trained our model using RNN and plotted the learning curve based on the stored output learning parameters in the memory. We have evaluated the model using the LIME technique of Explaination AI.

In our paper, We have proposed a Bi-LSTM model that outperforms some other machine learning models like Bernoulli's Naive Bayes , Linear support vector classifier and Logistic Regression.We have analyzed the data collected from Twitter in order to conduct an analysis of the sentiments behind the texts of corresponding twitter users using LIME technique of Explainable AI.

2 Related Works

Explainable AI is a class of techniques or methods that facilitates us to comprehend and evaluate a machine learning model's reached conclusion. We may use it to improve the performance of our models, as well as to guide others to understand how and what made the model take a particular decision. Some of the techniques belonging to that class are SHAP, DeepSHAP, DeepLIFT, CXplain, LIME. We integrated the LIME method with our created model as it makes a model's reaching conclusion individually comprehensive. Interpretations for different models and classifiers are all endorsed by LIME.

The major source of data for assessing human attitudes had been social media websites. H. Takkar et al. [4] conducted an investigation of several sentiment analysis techniques on the microblogging site twitter.They evaluated numerous lexical and machine learning techniques to sentiment analysis in their work. According to them, hybrid approaches provided superior results, with class two naive bayes being the optimal hybrid technique. They used the same dataset we have used.

Sentiment analysis necessitates the discovery of factual texts, since they lack both positive and negative sentiment. A research did by I. Chaturvedi et al. [5] to determine the difference between views and facts during sentiment analysis. They tested their algorithm on a variety of benchmarks, including twitter, where it achieved a score of more than 62 percent accuracy in detecting neutral mood in 498 tweets.

M. E. Mowlaei et al. [6] analyzed user comments on several e-commerce sites. They presented two changes to the previous aspect-based methodologies, statistical methods and genetic algorithm, respectively. On Bing Liu's customer

review datasets, it exceeds their prior work's t-test result and boosts accuracy, recall, and F-measure by 6.0, 1.0, and 7.4, respectively.

A comparision of sentiment analysers for twitter accounts using machine learning was conducted by A. Hasan et al. [7]. They examined the sentiment analysis algorithms TextBlob, SentiWordNet, and W-WSD using the Naive Bayes classifier. They demonstrated that W-WSD had the greatest accuracy (79 percent) on 6250 tweets. Additionally, they discovered that W-WSD achieved the maximum accuracy (62 percent) on 5000 tweets when employing an SVM classifier.

K. h. manguri et al. used the tweepy and TextBlob packages to analyze user data [8]. They gathered data from twitter using two hashtags: 'COVID-19' and 'coronavirus'. The data was acquired during one of the coronavirus's most active weeks, and they analyzed over 53 thousand tweets.

Sentiment evalution during critical occurrences using five classifiers had been done by G. a. ruz et al. [9]. They employed two Spanish datasets for the study: the 2010 Chilean earthquake and the Catalan independence referendum (2017). They concluded that their behavior was same regardless of language. SVM achieved the best accuracy on the first dataset, whereas Random forest achieved the highest accuracy on the second. Additionally, they said that TAN and BF TAN provide intriguing qualitative data that may be used to appreciate the major characteristics of an event's dynamic from a historical and sociological perspective.

K. sailunaz et al. created a recommendation system based on sentiment analysis of user tweets and responses [10]. They determined the agreement, sentiment, and emotion score of responses while calculating influence scores. Additionally, they compiled a list of tweet creators who had the same perspective on certain themes and expressed similar feelings and thoughts about those topics.

A study is performed by M. danilevsky et al.to examine the use of Explainable AI in natural language processing [11]. It summarized recent advances in XAI research in natural language processing as addressed at the main NLP conferences over the previous seven years. They discussed the major classification schemes for explanations, as well as the technical operations and explainability tools now available for developing explanations for model predictions.

S. m. mathews used Explainable AI to assess natural language processing on twitter data, tumor diagnosis on biological signals, and Windows pc malware detection using LIME [12]. They visualized their findings in an appropriate manner.

A framework is created by A. Waldis et al. for text features based on statistical information [13]. Their conclusion demonstrated their capacity to explain and forecast. They obtained superior results with the decision tree classifier than with the random forest and CNN classifiers.

The majority of these research used just two algorithms, used fewer tweets or data points, and achieved lower accuracy. We have worked with four models and 1.6 million tweets and achieved a better level of accuracy.

3 METHODOLOGY

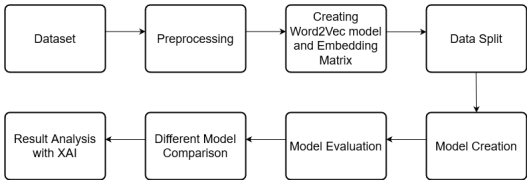

Fig. 1. Workflow Diagram

We have followed Fig. 1 for our work. The steps of this workflow are as follows:

3.1 Dataset

For our training dataset, we require only the sentiment and text fields, so we discard the rest. We have utilized the ISO-8859-1 encoding standard, which represents the first 256 uni-code characters. We have used the sentiment140 dataset consisting of 1.6 million tweets extracted using the Twitter API [3]. Initially, The tweets are labeled with 0 for annotating negative sentiments and 4 for annotating positive sentiments [3]. We remap the sentiment field with new values to reflect the sentiment. (0 = negative, 1 = positive).

The dataset contains the following 6 fields: sentiment (positive and negative), IDs, date, flag, user, and text [3]. We have plotted the distribution for the dataset to see whether we have an equal number of positive and negative tweets or not. As we can see from the Fig.2 , we have an equal number of positive and negative tweets. Both equal about eight hundred thousand tweets. This means our dataset is not skewed.

Note that, although the dataset description on Kaggle stated there were neutral tweets annotated as 2, No such annotation is present in the dataset.

3.2 Preprocessing

The tweets in our dataset contain some information apart from the text, e.g. user mentions, hashtags, urls, emojis, or symbols. Usually, NLP models cannot parse this non-text information. We need to clean up the tweet and replace tokens that actually contain meaningful information for the model.

We have utilized a list of words to execute some tasks like each text was changed to lowercase, Non-alphabets has removed by replacing numbers and alphabets with spaces are replaced by a predetermined dictionary

We also have employed regular expressions to define patterns, emojis, user name, urls and replaced them with proper labelings. We also have removed

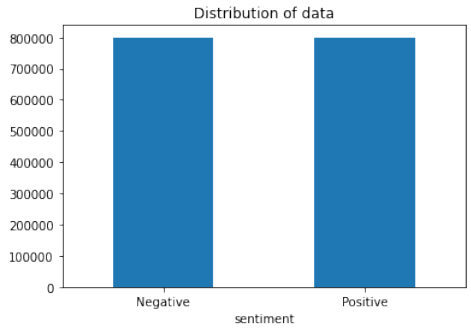

Fig. 2. Ratio of Positive and Negative tweets

non-alphanumeric and symbols and have added space on where needed utilizing regular expressions.

For tokenization, we have utilized TensorFlow Tokenizer and fitted the tokenizer on our training data for our tokenization process. Our input length is 60.

Since Bi-LSTM takes inputs of same length and dimensions, input sequences are padded into maximum length for all the tokens to meet the equal length. We have employed the 'pad_sequences' package from 'tensorflow.keras.preprocessing-.sequence' module for data padding to turn our sentences into sequences of tokens and pad them to make them of the same length. This also eliminates occurrences of the same character that have appeared in close proximity as the model will understand that the equal length padded words are similar in terms of context.

Neural network models need inputs to be of the same size. For that reason padding is utilized to make all the inputs of the same size. Padding is the process by which we can add padding tokens at the start or end of a sentence to increase it's length upto the required size, which is 60 in our case. Some words can also be dropped to fit the specified length.

3.3 Creating Word2Vec model and Embedding Matrix

Building and training effective word embeddings is a huge undertaking that takes millions of data samples and a lot of computing resources. In this case, we have built and trained an effective word embedding using an unsupervised learning technique called Word2Vec embedding. We use an embedding layer in our model to embed tokens into their vector representation. We get the embedding vocabulary from the tokenizer and the corresponding vectors from the embedding model. We set the embedded dimension to 100. The shape of the embedding matrix is usually the product of the vocab length and embedding dimension.

We then have created embedding matrix for all the words. Embedding matrix helps the model to recognize similar words having close values.

3.4 Data Split

Machine Learning models are trained and tested on different sets of data. We have divided pre-processed data into 3 sets to train, validation and test our model with the help of scikit learn:

- Training Data: The dataset upon which the model would be trained on, contains 80% data.
- Validation Data: The dataset upon which the model would be validated on, contains 10% data.
- Test Data: The dataset upon which the model would be tested against, contains 10% data.

3.5 Model Creation

We have adopted a hybrid architecture consisting of convolutional layer and Bi-LSTM layer. As mentioned earlier, Bi-LSTM cell state can pass contexts back and forth making the computation much more effective using the gates concept such as forget, input and output[11]. This resolves the RNN's Vanishing gradient problem (Weights reducing over time which leads to model's underfitting). Bi-LSTM with widely used in tasks like in natural language processing, voice recognition, image recognition and object detection as it resolves the long term dependencies.

4 Result Analysis

The following steps of Fig. 1 are the stages involved in the result analysis:

4.1 Model Evalution

With 80% training data, 10% validation data, and 10% test data, we have run our experiment on Bernoulli's NB [14], LinearSVC [15], Logistic Regression [16], and Bi-LSTM [17] model .

Bi-LSTM outperforms the other three models. As seen in Fig. 3, training accuracy of Bi-LSTM begins at 80% and increases to over 86% after 20 epoches. There, validation accuracy of Bi-LSTM starts at over 82% and rises to near about 85% after 20 epoches. After eight epoches, the accuracy of both becomes equivalent, and afterwards, the training accuracy exceeds the validation accuracy.

Such as in Fig. 4, training loss of Bi-LSTM starts at over 42% and drops to approximately 30% after 20 epoches. There, validation loss of Bi-LSTM begins at 38% and gradually declines to more than 34% after 20 epoches. After eight epoches, both types of loss become parallel, and the validation loss eventually outweighs the training loss. From the Fig 3 and Fig. 4 we can see that the data is not under-fitting, nor over-fitting, and is a good fit.

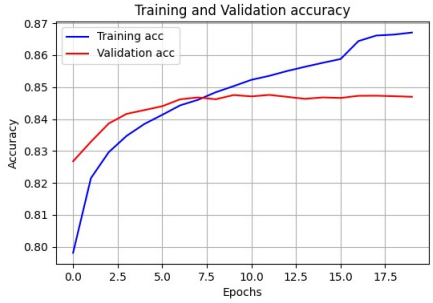

Fig. 3. Training and validation accuracy of Bi-LSTM

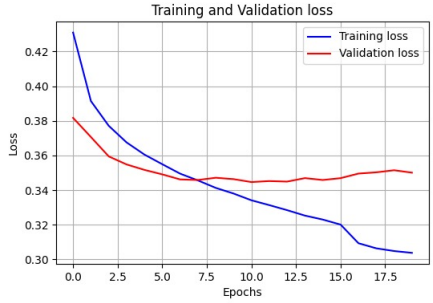

Fig. 4. Training and validation loss of Bi-LSTM

Table 1. Comparisons of other algorithms with our model

	Accuracy	**Precision**	**Recall**	**F1-score**
Bernoulli's NB	0.80	0.80	0.80	0.80
LinearSVC	0.82	0.82	0.82	0.82
Logistic Regression	0.83	0.83	0.83	0.83
Bi-LSTM	0.85	0.85	0.85	0.85

4.2 Different Model Comparison

According to TABLE 1, the bernoulli's naive bayes method has the lowest accuracy, precision, recall, and F1-score among the other models, although the linear support vector classifier algorithm and the logistic regression algorithm has practically identical scores. However, the accuracy, precision, recall, and F1-score of our suggested model Bi-LSTM are 0.85. Thus, it outperforms earlier models in terms of accuracy, precision, recall, and f1 score.

The confusion matrix for our suggested Bi-LSTM model is shown in Fig. 5. Here, the outcome is true positive for over 67000 tweets and true negative for

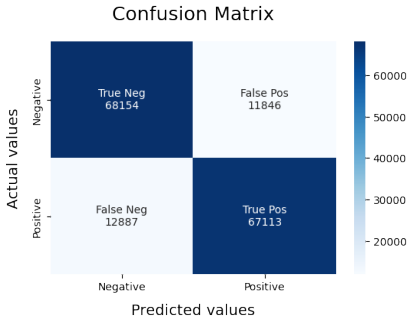

Fig. 5. Confusion Matrix of Bi-LSTM Model

over 68000 tweets. On the other hand, for almost 12000 tweets, the result is false positive, while for almost approximately 13000 tweets, the result is false negative.

4.3 Result Analysis with XAI

It is expected that an XAI model will be able to provide explanations for its findings while maintaining high accuracy and performance.Since we implemented XAI, our model is able to explain the reasons behind its reached conclusion based on the words in the sentence. We used the LIME method to explain the predictions. Some samples are shown on the later part.

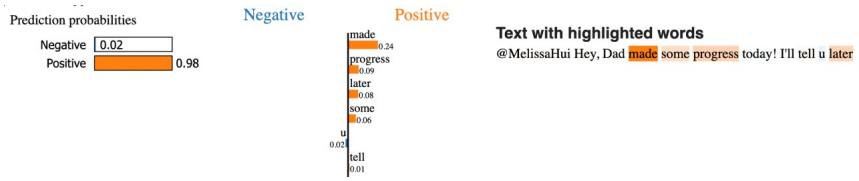

Fig. 6. Lime method for the tweet '@MelissaHui Hey, Dad made some progress today! I'll tell u later'

In the Fig. 6, it has classified the text '@MelissaHui Hey, Dad made some progress today! I'll tell u later' as positive based on the consisting words of that sentence. Note that the first line of the output is the text, the second line is the label, the third is the prediction that our model is making and the fourth one is the LIME output. The word 'made' is carrying most of the weight behind the prediction of the whole sentence's sentiment being positive.It is due to it being labeled as a positive word in the dataset that was used to train the model.

Fig. 7. Lime method for the tweet '@_alii nahhh she'll be cool i reckon that would be so awesomeeee haha chat tomorrowwwww ilyy'

Similarly, in Fig. 7, the word 'cool' in the tweet '@_alii nahhh she'll be cool i reckon that would be so awesomeeee haha chat tomorrowwwww ilyy' is carrying most of the weight of the prediction.

5 Conclusion

In this work, we have proposed a text-analyzing neural network model utilizing the Bi-LSTM RNN framework to detect depressive texts from among 1.6 million texts retrieved from the twitter social media platform. This algorithm has showcased 85% accuracy on this dataset which is more than other algorithms we have used. We also have implemented Explainable Artificial Intelligence (XAI) in our model that shows us the reason why our model classifies a text as positive or negative. We could have had achieved a better accuracy by doing further preprocessing of the data. Also, The dataset had few incorrect labels. we corrected the ones we noticed but to reach the best possible accuracy, corrections of all the labels of the entire dataset is required. So, better preprocessing of the dataset will give better results which will lead to greater accuracy. Implementing a hybrid deep learning model as well as exposure to larger dataset could also lead to a better accuracy. In the future, we plan to improve our own rnn model.

References

[1] R. ER, *Sentiment analysis: Sentiment analysis in natural language processing*, Sep. 2021. [Online]. Available: `https://www.analyticsvidhya.com/blog/2021/06/nlp-sentiment-analysis/`.

[2] B. Ghazaleh, R. Maciejewski, and H. Liu, *An overview of sentiment analysis in social media and its applications in disaster relief*, Apr. 2022. [Online]. Available: `https://link.springer.com/chapter/10.1007/978-3-319-30319-2_13`.

[3] M. M. KazAnova, *Sentiment140 dataset with 1.6 million tweets*, Sep. 2017. [Online]. Available: `https://www.kaggle.com/datasets/kazanova/sentiment140`.

[4] H. Thakkar and D. Patel, "Approaches for sentiment analysis on twitter: A state-of-art study," *arXiv preprint arXiv:1512.01043*, 2015.

[5] I. Chaturvedi, E. Cambria, R. E. Welsch, and F. Herrera, "Distinguishing between facts and opinions for sentiment analysis: Survey and challenges," *Information Fusion*, vol. 44, pp. 65–77, 2018.

[6] M. E. Mowlaei, M. S. Abadeh, and H. Keshavarz, "Aspect-based sentiment analysis using adaptive aspect-based lexicons," *Expert Systems with Applications*, vol. 148, p. 113 234, 2020.

[7] A. Hasan, S. Moin, A. Karim, and S. Shamshirband, "Machine learning-based sentiment analysis for twitter accounts," *Mathematical and Computational Applications*, vol. 23, no. 1, p. 11, 2018.

[8] K. H. Manguri, R. N. Ramadhan, and P. R. M. Amin, "Twitter sentiment analysis on worldwide covid-19 outbreaks," *Kurdistan Journal of Applied Research*, pp. 54–65, 2020.

[9] G. A. Ruz, P. A. Henríquez, and A. Mascareño, "Sentiment analysis of twitter data during critical events through bayesian networks classifiers," *Future Generation Computer Systems*, vol. 106, pp. 92–104, 2020.

[10] K. Sailunaz and R. Alhajj, "Emotion and sentiment analysis from twitter text," *Journal of Computational Science*, vol. 36, p. 101 003, 2019.

[11] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen, "A survey of the state of explainable ai for natural language processing," *arXiv preprint arXiv:2010.00711*, 2020.

[12] S. M. Mathews, "Explainable artificial intelligence applications in nlp, biomedical, and malware classification: A literature review," in *Intelligent computing-proceedings of the computing conference*, Springer, 2019, pp. 1269–1292.

[13] A. Waldis, L. Mazzola, and A. Denzler, "Towards explainable ai in text features engineering for concept recognition," in *International Conference on Statistical Language and Speech Processing*, Springer, 2020, pp. 122–133.

[14] H. Ismail, S. Harous, and B. Belkhouche, "A comparative analysis of machine learning classifiers for twitter sentiment analysis.," *Res. Comput. Sci.*, vol. 110, pp. 71–83, 2016.

[15] D. Effrosynidis, S. Symeonidis, and A. Arampatzis, "A comparison of pre-processing techniques for twitter sentiment analysis," in *International Conference on Theory and Practice of Digital Libraries*, Springer, 2017, pp. 394–406.

[16] T. Zahra, H. Ghous, and I. Hussain, "Sentiment analysis of twitter dataset using lle and classification methods," *International Research Journal Of Modernization In Engineering Technology And Science*, vol. 3, no. 1, pp. 1151–1164, 2021.

[17] R. Monika, S. Deivalakshmi, and B. Janet, "Sentiment analysis of us airlines tweets using lstm/rnn," in *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, IEEE, 2019, pp. 92–95.