Deep learning-based Sentiment Analysis of text

Ruth ONI
MSc. Artificial Intelligence and Data Science
4/1/2022

Table of Contents

Introduction

Sentiment analysis is one of the most actively researched areas of Natural Language processing (Zhang, et al., 2018). It differs from other areas of AI study because it deals directly with analyzing human sensibilities usually towards a subject. Sentiment analysis of text essentially describes the identification of sentiments in a text to generalize the polarity of opinions towards the subject. (Nasukawa & Yi, 2003).

Sentiment analysis matters greatly for its application across different spheres. It is inarguable that the success of any action may lie greatly on how informed the decision to take that action is. For individuals and organizations alike, there's great dependence on the availability of information about a choice regarding a product or service that affects its acceptance or pursuit. Thus, the opinion of other humans matters greatly in resulting human behavior. "Our beliefs and perceptions of reality, and the choices we make, are, to a considerable degree, conditioned upon how others see and evaluate the world. For this reason, whenever we need to make a decision, we often seek out the opinions of others" (Zhang, et al., 2018). The emergence and expansion of social media has enabled the generation of immense data related to products and services providing an opportunity to understand public opinions on a subject which can be relevant to policymaking, Business reforms, Product success, and even prediction of events. "These practical applications and industrial interests have provided strong motivations for research in sentiment analysis" (Zhang, et al., 2018). (Nasukawa & Yi, 2003) further highlight specific benefits including adaptation for "competitive analysis, marketing analysis, as well as risk management". Manual exploratory methods such as customer surveys and analysis exist to analyze reviews of a product, however at a great cost of time and resources (Ahmad, et al., 2021). These methods are limited and may be reductive because of the difficulty with asking the right questions to gather relevant opinions as well as the inadequacy of sample sizes (Nasukawa & Yi, 2003). Machine learning (ML) facilitates the computation of this process with its capacity to handle large data analysis and the use of predictive algorithms. Various ML approaches have been explored in the area of Sentiment analysis including Support Vector Machines (SVM), Naïve Bayes (NB), and Random Forest (RF) (Kurniasari & Setyanto, 2020). However, from a review of extensive research done in comparison with Deep Learning (DL), Neural Networks (NNs) perform better in classification tasks on the trade-off of large datasets for training (Dang, et al., 2021). This is because DL models are better at feature extraction than traditional ML algorithms as their architectural structure enables them to learn complex relationships and new abstractions within data at every layer.

This work, therefore, introduces Sentiment analysis with the use of Deep learning for opinion detection. The methods identify the sentiments of the Canvas Student mobile app and compare the results with a simple ML model. The next section will discuss sentiment analysis in review of related work. Section III will discuss the objectives of this work as well as the scope and methods of the experiments in this report, the outcomes and comparison of different approaches are finally presented in Section IV.

Section II

Background

According to (Nasukawa & Yi, 2003), The identification of sentiment is in three major components: “The Sentiment expressions, polarity and strength of the expressions, and their relationship to the subject”. The latter, which is commonly ignored in most applications, and the detection of the second component, being the most popular application. Polarity detection can be coarse-grained (i.e., positive, and negative) like this work undertakes or, “a fine-grained set of polarity categories (i.e., very negative, negative, neutral, positive, and very positive), which reflects the intensity of the polarity” (Rojas-Barahona, 2016).

DL has been applied to various Artificial Intelligence tasks including Image classification, robotics, climate, and business modeling. The application in Natural Language Processing (NLP) is most popular for semantic classification (Arnold, et al., 2011). The field of DL is facilitated by many powerful models encouraging research and its choice for sentiment analysis beyond lexicon-based models. The rest of this section briefly discusses related work to product user reviews classification with NNs.

(Moraes, et al., 2013) empirically compared SVM and Artificial Neural Networks (ANN) for document-level sentiment analysis. With feature selection and weighting in with Bag of Words for SVM. Their experimental results highlighted superior performance by the ANN in contexts of balanced and unbalanced datasets. The inference which they confirmed by hypothesis testing.

(Li, et al., 2014) Implemented a recursive neural deep model to classify reviews to polar sentiments in the Chinese sentiment treebank. The network predicted with an accuracy of 90.87%, outperforming three simple ML Algorithms including Maximum Entropy (87.46%), SVM (84.9%), and NB (78.65%).

(Chen, et al., 2016) adopted a hierarchical Long-Short Term Memory (LSTM) network to incorporate user and product information in addition to local text developing a User Product Attention-based NN model. The model was evaluated on three benchmark datasets across which it attained more than 4% improved performance when compared to simpler model baselines.

(Matsumoto, et al., 2020) proposed a Bidirectional-LSTM (Bi-LSTM) network with self-attention transformation and evaluated it on labeled and unlabeled data of mobile application reviews. The model was evaluated alongside pre-trained models, Star space and SVM based on a pre-trained model, BERT. The Bi-LSTM model outperformed all other models with a test accuracy of 80.1%.

(Poomka, et al., 2021) also compared the performance of ML models with DL in the analysis of reviews sentiment investigating the significance of various text pre-processing methods. They presented results highlighting that preprocessing is irrelevant to NNs but significantly improves the performance of ML models to even supersede NNs.

Most of the reviewed work analyzes benchmark datasets like IMDB, Yelp, and Chinese Treebank and compares them with existing baselines. This work provides labeled reviews based on the analysis of 40,000 reviews of the Canvas Student application collected from Google Playstore and Apple store. The labels consist of sentiment polarities (i.e., positive, negative, and neutral).

From reviewed literature, the most implemented NNs are Recurrent Neural Networks (RNN) especially LSTM-based ones as their structure enables the order of words and relevance of this order to be taken into consideration. Their results also point out the significance of large training data without preprocessing for adequate performance by these models. This work, therefore, proposes an LSTM network with architecture informed by the data.

The main experiment was carried out with the LSTM network including an embedding layer and was compared with a Bi-LSTM network as well as an RF Classifier.

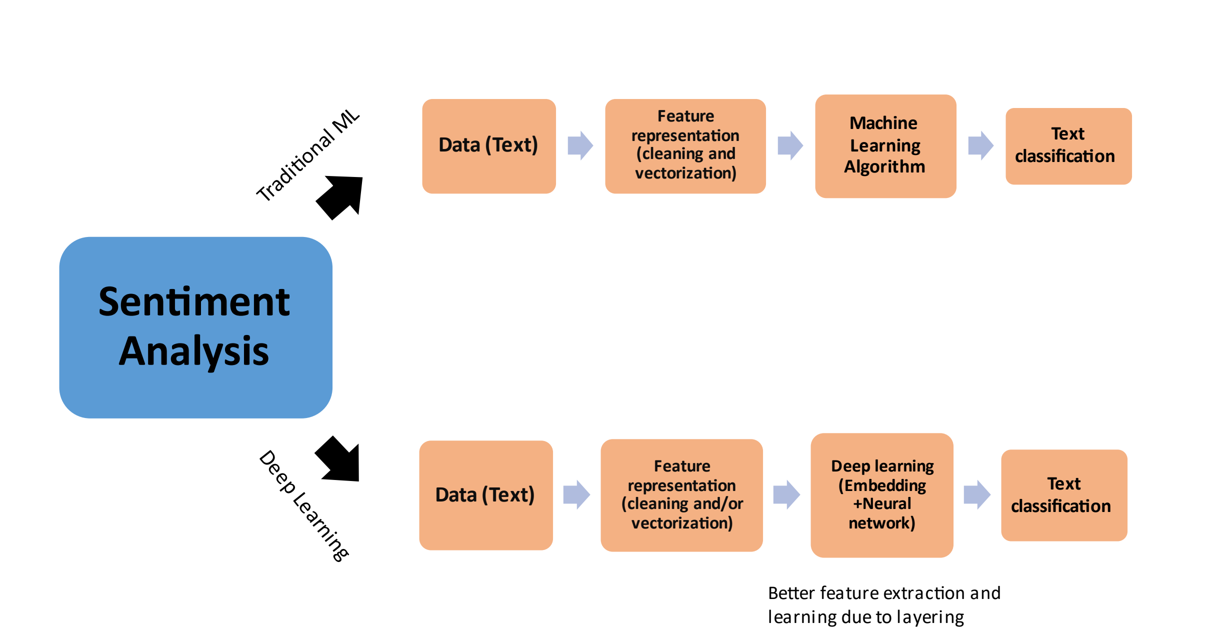


Figure 1: Sentiment Analysis frameworks; Simple Machine Learning vs Deep Learning.
Source: Author

Section III

Objectives

This work aims to evaluate the efficiency of Deep Learning in the analysis of sentiments of the Canvas student app.

The research objectives are aimed at determining if Neural Networks outperform Simpler models in the ML-based approach to sentiment analysis. The framework for the objectives is:

i. Data development
ii. Data preparation
iii. Define baseline – LSTM
iv. Implement Bi-LSTM
v. Implement Random Forest
vi. Evaluate results.

Methodology

Dataset Development Phase

In evaluating the DL models, the dataset of Android and IOS applications users' reviews for the Canvas app on Google play store and Apple IOS store was obtained using a third-party app "Appfollow" and annotated. All the reviews are in English Language and were obtained for the period December 20th, 2019, to March 27th, 2022. The data set collected comprises 18813 entries and 32 columns including ratings for the app. However, reviews are the prioritized feature for the analysis and experiments in this study. Before proceeding with the annotations, the data is also cleaned by removing unnecessary tokens, such as non-ASCII characters, and punctuations. The data was also lemmatized for context. The corpus was preprocessed only for label generation and one of three proposed classification tasks.

The data was analyzed, and labels were generated using the Lexicon-based (Yadav & Vishwakarma, 2020) NLTK Valence Aware Dictionary for sEntiment Reasoning (VADER) module; Sentiment Intensity Analyzer. The VADER was chosen because of its simple and computational economical approach as well as research-proven high level of accuracy (Hutto & Gilbert, 2015).

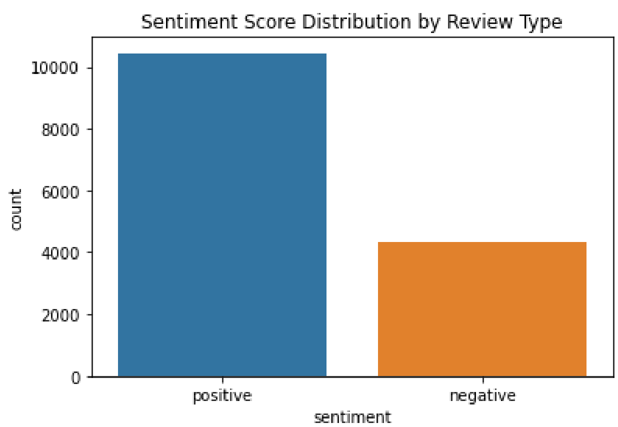


Figure 2 Label class distribution

Overall, the dataset provides an imbalanced number of samples per class. Fewer samples are obtained in the neutral category. The neutral class is not directly relevant to the objectives of this research and was therefore removed, reducing the number of entries to 14,787. Specifically, we have 10,444 and 4343 samples in the positive class and negative classes respectively. For the classification experiments, the data was divided into training and test sets with a proportion of 75 % and 25%, respectively.

Since both DL and classical ML were employed for the classification tasks. Data transformation for both Methods was carried out using Keras Tokenizer and CountVectorizer respectively while the sentiments were transformed for encoding using pandas dummies.

Experimental Phase

The main experiment in this work is the implementation of an LSTM network to classify reviews based on the Document/sentence level of sentiment analysis and to determine the polarity of the reviews to be Positive or Negative. The classification task is binary particularly due to fewer records of the neutral sentiment class and LSTM was chosen as it is the most commonly implemented deep learning model for sentiment analysis due to its temporal capabilities.

Recurrent Neural Networks (RNN)

We may consider reviews as a sequence of words whose order can be fundamental to predicting their sentiments. According to (Nicholson, 2020), RNNs are very powerful ANNs because of their ability to recognize patterns in sequences of various data types. They exceed their counterparts because of their temporal dimension, order of time, and sequence. Unlike feedforward Neural Networks (FNN), who as described by (Nicholson, 2020) are “amnesiacs of their recent past and only remember the formative moments of their training”, RNNs cycle their input in a loop such that a current input is taken into consideration forward in time. The addition of memory to their structure enables them to perform tasks FNNs cannot especially finding long-term dependencies between events separated by many moments (Nicholson, 2020).

The structure of RNNs are not completely abstract from ANNs. The nodes in a typical ANN are compressed into a single layer of RNNs (Biswal, 2022). The structure of a RNN is diagrammatically represented as:

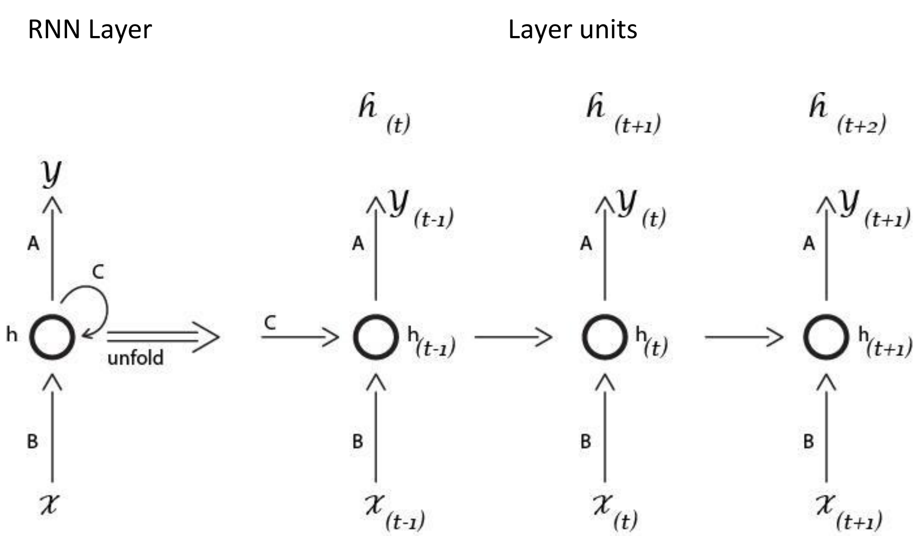


Figure 3.0 Structure of a RNN Layer
Source: Author

where **X** is the input layer, **h** is the hidden layer and **y** is the output, and **A**, **B**, and, **C** are parameters of the network (Singh, 2021). In describing RNNs as an extension of FNNs, it will be feeding the output at every step as input back to the network such that **X** at any given time t is **X(t)** and **x(t-1)** combined.

Mathematically, the current state of an RNN **h(t)** is (Singh, 2021) :

$\mathbf{h_{(t)} = fc\,(h_{(t-1)}, x_{(t)})}$ Where,

$\mathbf{h_{(t)}}$ is the current (hidden) state, $\mathbf{h_{(t-1)}}$, is the previous state and $\mathbf{x_{(t)}}$ is the current input.

With the addition of the activation function (Singh, 2021),

$\mathbf{h_{(t)} = tanh\,(w_{hh}\, h_{(t-1)} + w_{xh}\, x_{(t)})}$

The final output will be,

$$y_{(t)} = w_{hy} h_{(t)}$$

Here, $y_{(t)}$ is the output state, w_{hy}: weight at output layer, $h_{(t)}$: current state (Singh, 2021)

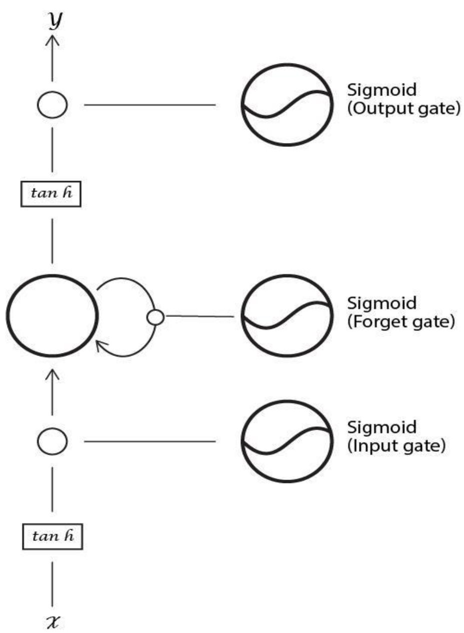


Figure 3.1 Structure of an LSTM unit
Source: Author

Long-Short Term Memory RNNs (LSTM)

Simple RNNs are however limited by the effects of their temporal nature in the backward propagation phase of the NNs. LSTM is not limited by the vanishing gradient problem and the exploding gradient problems that characterize RNNs. (Graves, et al., 2009). Its structure includes cell gates in addition to its original hidden state that summarily allows better control of the information that flows through the network and retains long-term dependencies (user:2444, 2021). The hidden layer includes memory cells with corresponding gate units, and it unfolds into the number of steps as dictated by the parameters of the network. This structure allows it to determine and forget unnecessary information that should not go back into the network as input (Sinha, 2019).

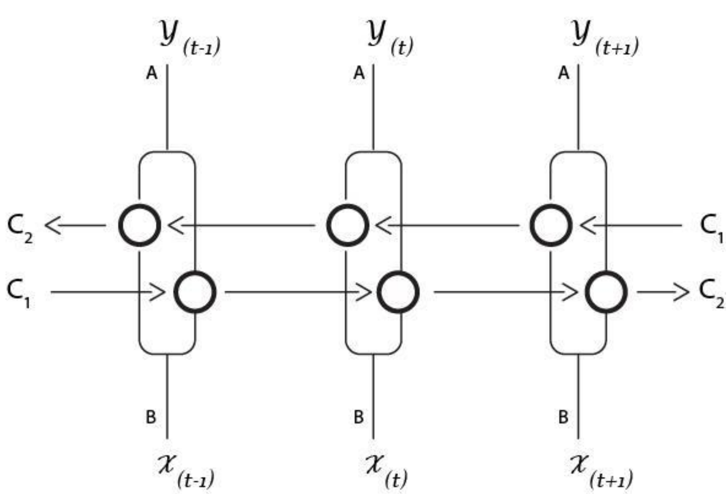


Figure 3.2 Structure of units in a BI-LSTM Layer
Source: Author

Bidirectional- LSTM (Bi-LSTM)

A second NN, Bi-LSTM is evaluated on the data set. “The basic idea of a bidirectional RNN is to present each training sequence forwards and backward to two separate recurrent nets, both of which are connected to the same output layer” (Graves & Schmidhuber, 2005). This implies, the network in comparison with a simple LSTM has more temporal information about every point in the network at every iteration in the sequence that is complete and sequential. The structure of a Bi-LSTM allows it to process data in two opposite directions (two hidden layers) with long-range context due to the LSTM content which is then output in the same sully connected layer (Graves, et al., 2013). The Bi-LSTM is an improvement on the LSTM Neural network (Brownlee, 2017).

Random Forest (RF)

RF is an ensemble learning method that selects features randomly from a sample to construct a collection of decision trees with controlled variation (Fawagreh, et al., 2014). Furthermore, Randomization is also applied in the selection of nodes to be split during the construction of a decision tree.

All three models will be evaluated using the Canvas student App reviews dataset and to measure the performance of the classifiers, we review the classification report (Recall, Precision, and Accuracy) from the ScikitLearn Metrics API and the accuracy score of the Keras network classification. This choice is confident to evaluate true and false positives since the dataset is not balanced.

Experiments

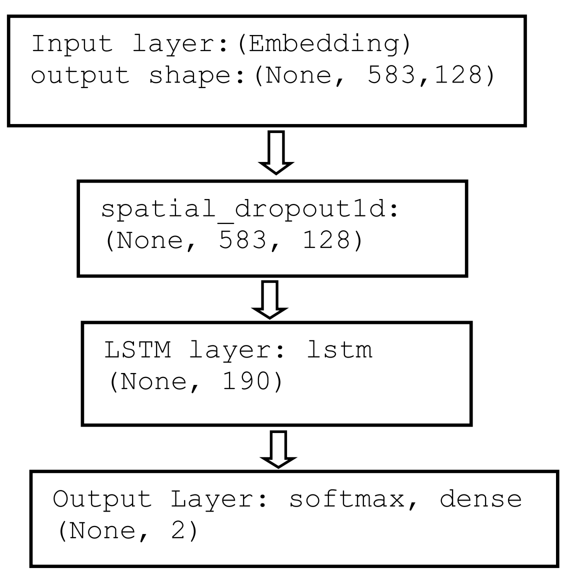
In the experiments to assess the sentiment classification performances of traditional ML versus DL algorithms, we apply two scenarios for performance evaluation; The development of classification models from non-preprocessed data and, the development of a model with preprocessed data (Poomka, et al., 2021). The ML algorithm for model creation is based on RF with CountVectorizer transformation. The DL algorithms are based on LSTM and Bi-LSTM with word embedding transformation.

As this work implements NNs, we do not train a vectorizer for the word embeddings, we instead include an embedding layer in the network for it to learn the word embeddings on its own (Chablani, 2017). However, the Tensor flow "Tokenizer" was used to transform the text and convert it into a sequence of integers with restrictions to use only the total unique number of words and pad sequences to convert the vectors to a 2D- array.

In preprocessing the text for RF, we remove prefix-suffix from words as well as stop words from the data. Following this process, the CountVectorizer module was applied for word embedding and vectorization. The data was then split into the training and test subsets.

Experimental setup-

Model Architecture:



From the embedding layer of the network, the new representations of the tokenized data will be imputed to the LSTM cells. "The number of parameters cannot be greater than the number of examples at the risk of overfitting" (Chablani, 2017), the input vector is therefore defined according to the input dimension of each word value. The LSTM layer has its number of hidden neurons and its input sequence transformed into a vector sequence size 190 (lstm_out) similar to implementation by (Sinha, 2018). A dropout layer including recurrent dropout with a 0.2 dropout rate was added to prevent overfitting. The output layer of the network is a 2-unit dense layer with a softmax activation function because it is fast and less prone to saturation. The model is then compiled with Adam because of its default learning rate and categorical cross-entropy for the loss due to the nature of the classification task. Similar architectural parameters are maintained for Bi-LSTM, and both were trained for 10 epochs. For RF, we fit the model with default hyperparameters except for 150 n_estimators and a random state of 101.

As part of the experiment, the Models are validated with the open test corpus (test dataset), derived from the original benchmark, and reviewed based on the classification report.

Section IV

Results

Both Neural networks achieved test accuracy of 92% during training (see figures 4.0 & 4.1)

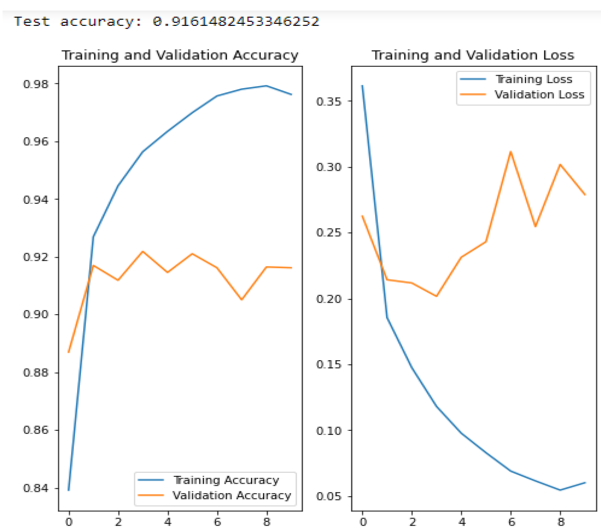


Figure 4.0 Network Accuracy and Loss during training; LSTM

Figure 4.1 Network Accuracy and Loss during training; Bi-LSTM

Table 1. presents the evaluating metrics of all three model predictions.

Model	Accuracy (%)	Precision (%)	Recall (%)
RF + CountVectorizer	**92**	**93**	**96**
LSTM	**92**	**95**	**93**
BI-LSTM	**95**	**93**	**95**

Table 1 Accuracy, Recall, and Precision scores of the Random Forest, LSTM, and Bi-LSTM models

The final models are presented in table 1. The overall observation is that there are no major differences between the models, in terms of Precision, LSTM achieved a score of 95%, which is just slightly higher than the other models including RF.

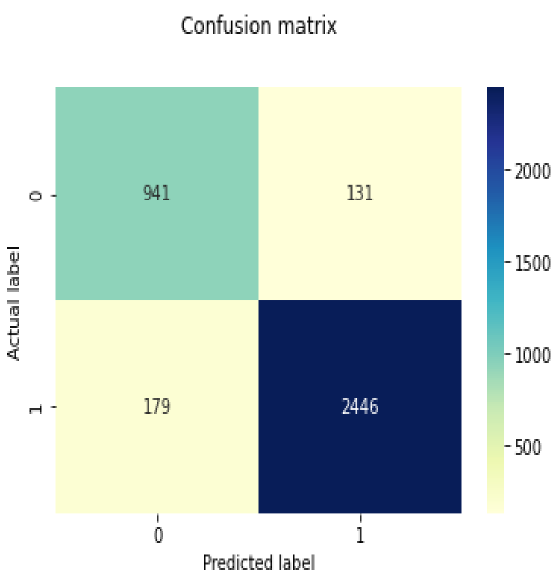


Figure 5.0 True positives and Negatives of the LSTM model's predictions

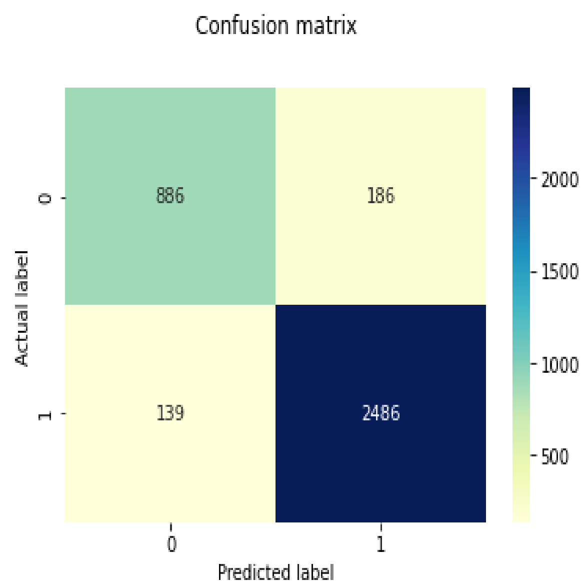


Figure 5.1 True positives and negatives of the BI LSTM model's predictions

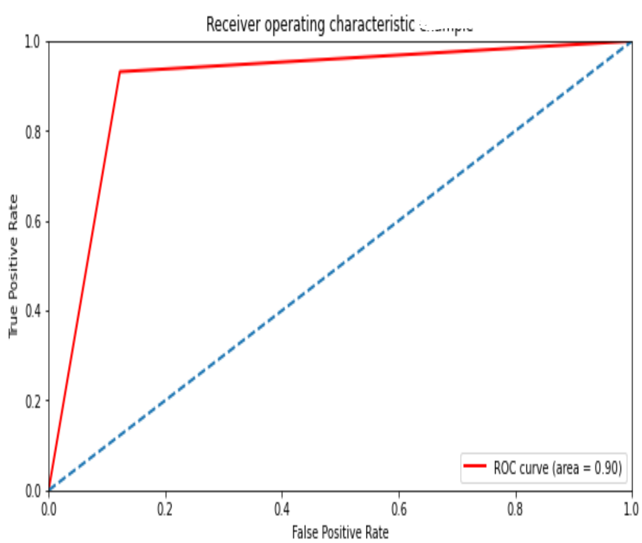


Figure 6.0 Receiver operating curve for LSTM.

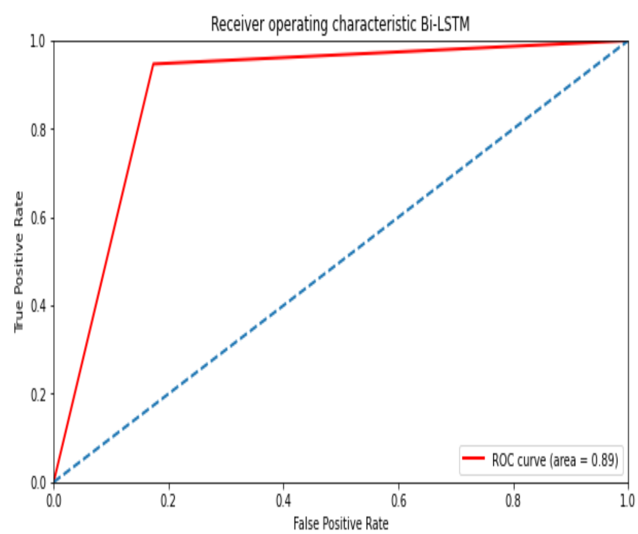


Figure 6.1 Receiver operating curve for Bi-LSTM

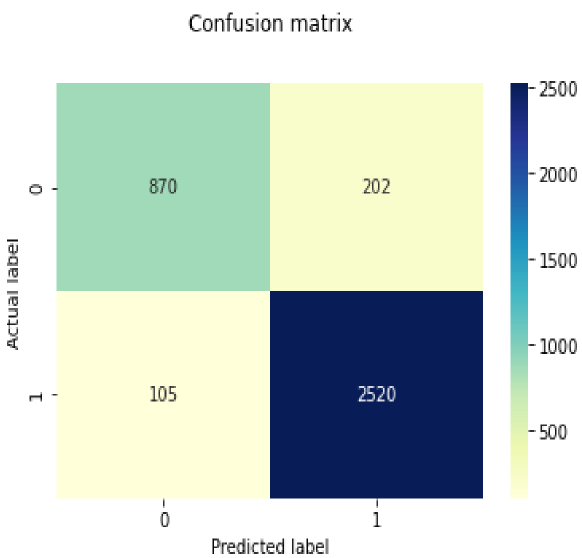


Figure 7.0 True positives and Negatives of the RF model's predictions

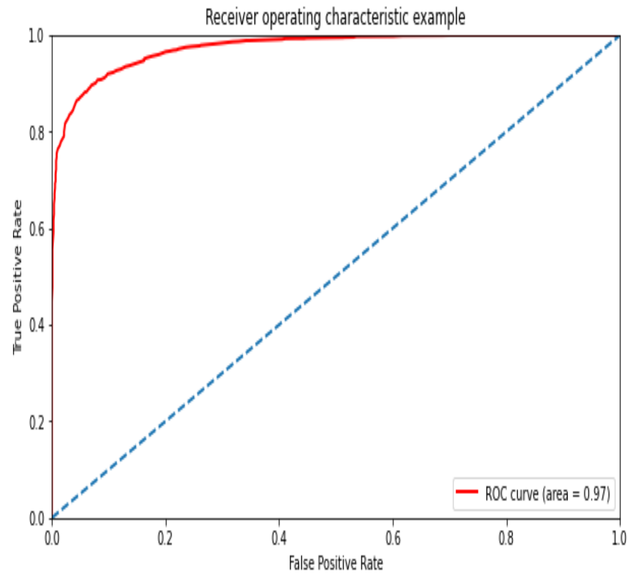


Figure 7.1 Receiver operating curve for RF

The trade-off between the true positive (TP) and false positive (FP) rate is shown in the Receiver Operating Characteristics (ROC) curve and can be used to access the quality of the classifier used in a model. The distance between the ROC curve and the diagonal baseline indicates the reliability of the predictions from our model. RF outperforms the NNs with an area under curve (AUC) value of 0.97 (See figures 5.0 to 7.1).

Conclusion

This paper reviewed the classic ML and DL applications and approaches to the task of sentiment analysis. The benefits and limits of both approaches used in sentiment analysis have been reviewed and performances of the classification models have been compared. The classification report of the models based on the Canvas App Reviews dataset illustrated that ML approaches such as Random Forest can perform with similar results as Neural Networks with the right transformations for text. The training time complexity of the models was however dissimilar, which favors RF with the NNs being computationally very expensive. This study suffers from limitations (imbalanced data and computational power to review and optimize NNs), the recommendation is therefore to begin with Random Forest when presented with similar situations. The insights from this study may be used as a basis for further studies. Including investigating the performance of both for other NLP tasks or the application of less computationally expensive NNs like Gated Recurrent Units for sentiment Analysis. The influence of context and preprocessing in expectation of better performance from NNs can also be studied.

References

Ahmad, K. et al., 2021. *Sentiment Analysis of Users' Reviews on COVID-19 Contact Tracing Apps with a Benchmark Dataset.* [Online]
Available at: https://doi.org/10.48550/arXiv.2103.01196
[Accessed 6 March 2022].

Arnold, L., Rebecchi, S., Chevallier, S. & Paugam-Moisy, H., 2011. *An Introduction to Deep Learning.* Bruges, s.n.

Biswal, A., 2022. *Recurrent Neural Network (RNN) Tutorial: Types, Examples, LSTM and More.* [Online]
Available at: https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn
[Accessed 26 March 2022].

Brownlee, J., 2017. *A Gentle Introduction to Long Short-Term Memory Networks by the Experts.* [Online]
Available at: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/
[Accessed 14 April 2022].

Chablani, M., 2017. *Sentiment analysis using RNNs(LSTM).* [Online]
Available at: https://towardsdatascience.com/sentiment-analysis-using-rnns-lstm-60871fa6aeba
[Accessed 26 03 2022].

Chen, H. et al., 2016. *Neural Sentiment Classification with User and Product Attention..* Austin, Association for Computational Linguistics.

Dang, C. N., Moreno-García, M. N. & De la Prieta, F., 2021. Hybrid Deep Learning Models for Sentiment Analysis. *Complexity,* 2021(9986920), pp. 1-16.

Eckhardt, K., 2018. *Choosing the right Hyperparameters for a simple LSTM using Keras.* [Online]
Available at: https://towardsdatascience.com/choosing-the-right-hyperparameters-for-a-simple-lstm-using-keras-f8e9ed76f046
[Accessed 14 April 2022].

Fawagreh, K., Gaber, M. M. & Elyan, E., 2014. Random forests: from early developments to recent advancements. *Systems Science & Control Engineering,* 2(1), pp. 602-609.

Graves, A. et al., 2009. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 31(5), pp. 855-868.

Graves, A., Mohamed, A.-r. & Hinton, G., 2013. *Speech recognition with deep recurrent neural networks.* Vancouver, IEEE .

Graves, A. & Schmidhuber, J., 2005. *Framewise phoneme classification with bidirectional LSTM networks.* Montreal, IEEE International Joint Conference on Neural Networks.

Hutto, C. J. & Gilbert, E., 2015. *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text.* Ann Arbor, s.n.

Kurniasari, L. & Setyanto, A., 2020. *Sentiment Analysis using Recurrent Neural.* Bukittinggi, IOP Publishing.

Li, C. et al., 2014. *Recursive Deep Learning for Sentiment Analysis over Social Data.* Washington, IEEE Computer Society.

Matsumoto, K. et al., 2020. classification of Smartphone Application Reviews UsingSmall Corpus Based on Bidirectional LSTM Transformer. *International Journal of Machine Learning and Computing,* 10(1).

Moraes, R., Valiati, J. F. & GaviãO Neto, W. P., 2013. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications: An International Journal,* 40(2), p. 621–633.

Nasukawa, T. & Yi, J., 2003. *Sentiment Analysis: Capturing Favorability Using Natural Language Processing.* Sanibel Island, FL, USA , Association for Computing MachineryNew YorkNYUnited States, pp. 21 -.

Nicholson, C., 2020. *A Beginner's Guide to LSTMs and Recurrent Neural Networks.* [Online]
Available at: https://wiki.pathmind.com/lstm
[Accessed 14 April 2022].

Poomka, P., Kerdprasop, N. & Kerdprasop, K., 2021. Machine Learning Versus Deep Learning Performances on the Sentiment Analysis of Product Reviews. *International Journal of Machine Learning and Computing,* Volume 11, pp. 103-109.

Rojas-Barahona, L. M., 2016. Deep learning for sentiment analysis. *LangLinguist Compass,* Volume 10, pp. 701-719.

Singh, A., 2021. *Recurrent Neural Networks : Introduction for Beginners.* [Online]
Available at: https://www.analyticsvidhya.com/blog/2021/06/recurrent-neural-networks-introduction-for-beginners/
[Accessed 26 March 2022].

Sinha, N., 2018. *Understanding LSTM and its quick implementation for Sentiment Analysis.* [Online]
Available at: https://towardsdatascience.com/understanding-lstm-and-its-quick-implementation-in-keras-for-sentiment-analysis-af410fd85b47
[Accessed 21 04 2022].

Sinha, N., 2019. *Understanding LSTM and its quick implementation in keras for sentiment analysis.* [Online]
Available at: https://towardsdatascience.com/understanding-lstm-and-its-quick-implementation-in-keras-for-sentiment-analysis-af410fd85b47
[Accessed 26 March 2022].

user:2444, 2021. *What is the difference between LSTM and RNN?.* [Online]
Available at: https://ai.stackexchange.com/questions/18198/what-is-the-difference-between-lstm-and-rnn
[Accessed 26 March 2021].

Yadav, A. & Vishwakarma, D. K., 2020. Sentiment analysis using deep learning; a review. *Artificial Intelligence Review,* Volume 53, pp. 4335-4385.

Zhang, L., Wang, S. & Liu, B., 2018. Deep Learning for Sentiment Analysis. *WIREs Data Mining knowledge Discovery,* 8(1253), pp. 1-25.