Comparative Analysis of RNN, LSTM and BERT for Sentiment Analysis on Twitter Dataset

Siyam Sajnan Chowdhury
School of Computer Science
University of Windsor
Windsor, Canada
chowdh1b@uwindsor.ca

Kasra Mojallal
School of Computer Science
University of Windsor
Windsor, Canada
mojalla@uwindsor.ca

Kimia Tahayori
School of Computer Science
University of Windsor
Windsor, Canada
tahayor@uwindsor.ca

***Abstract*—Sentiment analysis is a subset of Natural Language Processing (NLP) and it is used to mine information on sentiments, emotions or perspectives on a specific domain from textual data which is considered to be subjective [1]. Sentiment Analysis is also known as opinion mining. NLP serves a wide range of uses and purposes. When the idea of NLP first stemmed, the use of Recurrent Neural Network (RNN) as the primary model was widespread and prevalent [2]. Later on, with the development of Long Short-Term Memory (LSTM) [3], NLP tasks started employing it as it provided performances improvements over the traditional RNN model. In recent times, the discovery of the transformer architecture [4], especially Bidirectional Encoder Representation from Transformers (BERT) [5] brought about a significant performance improvement in sentiment analysis tasks. In this paper, we have tested and compared the performance of the three models on the Twitter 140 dataset [6] and simulated the performance improved of BERT over traditional models such as RNN and LSTM.**

Keywords: Sentiment Analysis, Natural Language Processing, RNN, LSTM, Transformers, BERT.

I. INTRODUCTION

NLP has been expanding for the last five years [7], and it has seen large and widescale implementations such as ChatGPT, which uses the pretrained model GPT4 [8], being one of the biggest applications of it which garnered over a 100 million users in less than a year [9]. This illustrates the tremendous potential of NLP.

Sentiment Analysis is a subset of NLP and it is used for a wide range of applications. Sentiment analysis is used to extract sentiments and emotions from data that is provided in text form. This doctrine of extracting sentiments from a person's text data has huge implications both commercially as well as in research. It could be used in the phase of product development where people's feedback is used to determine their perception of the product which in turn can be used to improve upon the product [10]. It could also be used to determine a person's state of mind which could in turn be used to help with their mental health diagnosis and prescription [11].

There are various discrete levels of sentiment that the extracted sentiment can be classified into but for the purposes of this paper, we used the labels positive and negative as per the labels in the Twitter 140 dataset.

RNN and LSTM were the key models that were used to classify sentiments a decade and a half back from now but since the discovery of transformer architecture in 2017, it has been used quite extensively for NLP and sentiment analysis tasks.

In 2018, with the discovery of Universal Language Model Fine-tuning for Text Classification (ULMFiT) [12], the concept of transfer learning was implemented for natural language processing tasks. Multiple NLP and Sentiment Analysis tasks now employ this learning method to train language models. This can be illustrated by the fact that numerous large-scale transformer-based language models are currently in use with more improved and updated ones on the way. Some of the key language models based on the transformer architecture are as follows:

Megatron – Nvidia [13]
GPT-4 – OpenAI [14]
BERT – Google Research [5]
Turing-NLG – Microsoft [15]

II. PROBLEM STATEMENT

Conventional models which were used for sentiment analysis had issues with efficiency and comprehensibility of contexts. One of the prime problems of those models were that they overlooked considering neighboring words and sentences while considering individual words as independent parts or units. This was a huge flaw as it fails to take into account the impact surrounding words could have on determining the meaning of a word. This is especially important in case of words which are sarcastic or ironic in nature. If a specific word is considered without the rest of the words in the sentence in a sentence which is implying irony or sarcasm, it could extract a sentiment totally different from the one meant by that sentence. This is why comprehending contexts is so important.

Traditional models also require fully labelled models which can be quite expensive to create both in terms of time and money as it would most definitely require a lot of human input in comprehending and labelling. This can still work well with widely used languages such as English or Mandarin but for languages which are less popular such as Persian or Hebrew, it can negatively affect due to the lack of datasets which are properly labelled. User generated data on social networking sites could also contain words that are not contained in the dictionary as well as variations of misspellings of a word, abbreviations as well as urban slangs. These hinder the correctness and accuracy of a model's performance.

Conventional models have issues with relying too much on methods which are based on considering lexicons only. This means that these models seek out words which are contained in a pre-trained and pre-modelled list of special words which are sentiment deriving in nature. Overemphasizing on certain words could also mean that words implying negation such as "not" could be overlooked turning the phrase "not good" to be classified as positive.

Use of RNN over traditional models reduced a good number of this issues but many other issues persisted. One of the biggest issue with RNN was the vanishing gradient problem which fails to accurately capture the sentiment from long strings of texts. In long strings of data, the impact of the words at the start of the text becomes increasingly smaller to a point where the changes to the parameters based on the gradient becomes so nominal that it becomes almost negligible. This is why it is called a vanishing gradient problem.

An issue associated with this is RNNs only take into consideration the previous state while factoring the output state based on the input state due to its sequential input. This means that it would make it impossible to accurately capture the global context as sequential data input means that there is no way for it to know what later words could imply and possibly affect the meaning of a sentence. For example, if there are two sentences where the first sentence states that something is not good, but the second sentences exclaims that the first sentence was a joke and that was indeed very good could be caught by RNN as a negative or a neutral sentiment. This is the limitation of capturing the global context of RNN.

Sequential input of RNN means that it lacks the capacity to capture inputs in parallel. This is a drawback when working with big data as it will be very slow to train thus driving up the costs. It also means that it would not be able to take advantage of modern processing technology of GPUs which are made specifically to be able to do parallel processing. This inability to use modern technology renders the use of RNN for sentiment analysis tasks quite slow.

RNNs also have difficulties in case of variable length strings. It can only take in fixed length sequence as sequential input so it causes longer strings or sequences to be cut off or truncated which could result in the loss of precious sentiment bearing information. This could drive down the model's performance significantly.

LSTM overcomes a lot of these issues that RNN had chiefly the ability to take in longer strings by reducing the effect of vanishing gradient by capturing long-range dependencies. LSTM's memory states enable it to work with sequences which have different lengths thus requiring no truncation or deletion of important information. LSTM has better comprehension capability when it comes to contexts and negations. However, it still falls short on the ability to make use of parallel processing.

BERT, which is made from the transformer architecture has a significant advantage over the other two models because it can take all the input data at once. This means that it can map and model the relationship between the different words using the attention mechanism and have a good understanding of the global context. Taking data all at once also means that it can make use of modern-day computing technologies such as the parallel processing units of modern-day GPUs making it very fast and resourceful.

It has the added benefit of making use of pre-training and fine-tuning. Pre-training is a part of transfer learning where a model is already fed a lot of data for it to understand the language which enables it to have comprehension of different types of words that are not contained in conventional dictionaries.

This paper seeks to employ BERT as a solution to the issues that RNN and LSTM had and to simulate how well it performs compared to the models RNN and LSTM and to question whether the use of BERT over these models is justified.

III. RELATED WORKS

Some papers that we considered while working on this paper are as follows:

Delvin et al. first discovered and used BERT in their paper where they extracted the encoder from the transformer architecture and used it on eleven NLP tasks and found state-of-the-art results in the domain of language processing by using an already pre-trained model which they later fit separately for the different tasks. [5]

Hoang et al. applied BERT and created a sentiment analysis model based on aspects, by fine-tuning the model to a sentence pair classification model. Their model outperformed previous state-of-the-art aspect-based sentiment analysis models [16].

Zhang et al. used bidirectional LSTM in GNN for the embedding for sentiment analysis [17].

Zimbra at al. ascertained the mean accuracy for generalized sentiment analysis tasks were 61% and specialized tasks were about 72% [18].

IV. METHODOLOGY

A. Material and Data

We used the Twitter 140 dataset [6] in order to analyze the comparative performance of the 3 models that we used – RNN, LSTM and BERT.

The Twitter 140 dataset has 1.6 million data points with the target labels being 0 and 1 – 0 standing for negative sentiments and 1 for positive sentiments. For our paper, we used 7000 data for training and 1000 data for validation and 2000 data for testing due to constraints in processing capacities.

B. Proposed Models

I. Recurrent Neural Network

RNN, which stands for Recurrent Neural Network, is a neural network which, unlike the single direction of movement of traditional feedforward neural networks, can move in loops within a unit. By making use of its contextual comprehension of temporal dependencies, it works really well on time series data which is a sequential data model. [19]

The core functional unit of an RNN is the recurrent unit. This is a hidden state which works as the cell providing it with memory. RNN uses the input and the output from the previous state to produce an output.

The unit on the left of figure 1 is a single RNN unit. This, when stacked multiple times, forms the different layers of an RNN model. The input is the x(t), h(t) is the hidden layer which is computed at every time step as any

predetermined non-linear function such as ReLU (Rectified Linear Unit) [20] applied on the input x(t) as well as the hidden state of the previous time step h(t-1) and o(t) is the output. U is the connection between x(t) and h(t) and W is the weight between the h(t) and h(t-1) and V is the parameter between h(t) and o(t). These parameters U, V and W are weight vectors which are updated using backpropagation with every iteration thus learning the dependencies and patterns.

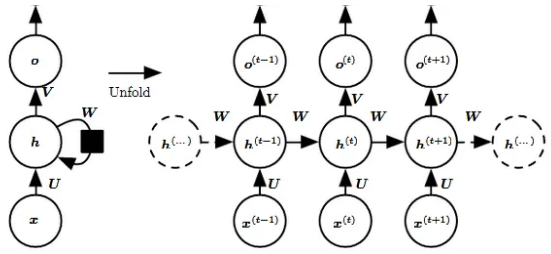

Figure 1: RNN ARCHITECTURE [21]

II. Long Short-Term Memory

LSTM has a lot of similarity in terms of structure to the RNN as it is based on the RNN model. It solves the vanishing gradient problem that RNNs typically suffers from by comprehending the long range dependencies comparatively better. This is done via memory cells which can choose which information to retain and which ones to forget and by doing this, it could choose to retain information from earlier iterations thus minimizing the diminishing gradient of the loss function.

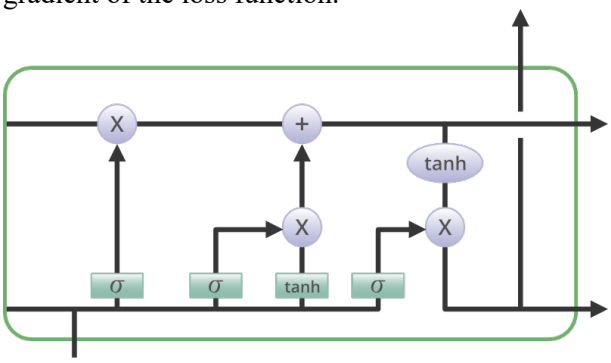

Figure 2: LSTM ARCHITECTURE [22]

LSTM has three gates - Input Gate, Forget Gate, and Output Gate.

The input gate has two activation functions – sigmoid and tanh. The task of the input gate is to determine the values that are to be used to update the memory cells and which values to discard. The sigmoid function acts as the passing through gate which chooses which values will pass and the tanh function assigns weights based on importance to those values that pass and they range from -1 to 1 [23].

Forget gate is quite binary in nature as it uses the sigmoid function to choose whether to remember or forget a value with the values being binary – 0 or 1. [23]

The output gate is similar to the input gate with a sigmoid and tanh function. The sigmoid function chooses which values are to be passed through as output based on the input and the hidden cell and tanh is then used to append weight to the values that passes through the sigmoid function by determining the product of the two functions [24].

III. BERT

Before explaining the concept of BERT, understanding the concept of Transformers is required. Transformers solves seq-to-seq problems by employing self-attention mechanism [25]. The transformer architecture consists of two parts – the encoder and the decoder, as seen in figure 4. BERT is actually taking the encoder from the transformer model and stacking them up together end to end as seen in figure 3.

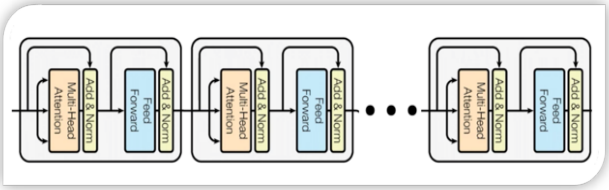

Figure 3: BERT ARCHITECTURE [5]

The encoder of the transformer architecture is quite simple as it consists of just two key parts – the multi headed attention and the feed forward neural network. Input is provided to the encoder model which implements multiple layers of self-attention mechanism to create individual attention vector for each word which are then fed to the feedforward neural network. The output vector from the feedforward neural network is each token's contextualized representation.

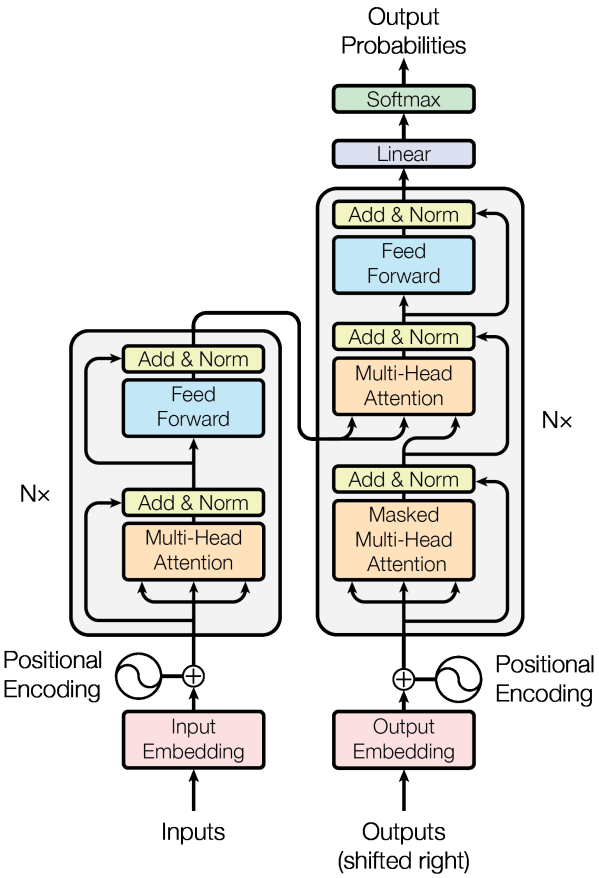

Figure 4: TRANSFORMER ARCHITECTURE [23]

In order to train BERT before giving the input to it, a set percentage of words are exchanged with masked token which BERT tries to figure out from the surrounding words and context. This is done by adding a classification layer at each input, embedding it into a vocabulary dimension and provide a prediction of the probability of every single word in that vocabulary using the softmax function. The convergence of the model is slow as the loss function is based on the accuracy of the unmasked values.

We applied transfer learning to the pre-trained BERT model so that it already came with an understanding of the language. We trained the pre-trained BERT model on our dataset which is also known as fine-tuning, which is done to give the model context of what it is to predict [26]. Using a pre-trained model instead of training the model from scratch was because it would require huge computational resources and time.

The application of transfer learning in this algorithm BERT is by fine-tuning the BERT model to our own dataset. We fine-tuned our data for the BERT model which means we trained the pre-trained data further on our dataset so that it adapts to the context of our dataset and performs accordingly by understanding the context [17]. Besides, training a huge model like BERT from scratch would take a very long time and huge computational requirements to train.

C. Preprocessing

Preprocessing is done to convert the dataset to a format that is workable by the model and can be analyzed and interpreted. In order to remove different meaningless special characters, conjunctions, and words such as prepositions (to, in, on) and articles (a, an, the) which added absolutely nothing of importance or significance while extracting the sentiments, we used the NLTK library [27]. We used it for stop-words removal, punctuation removal, lemmatization, and stemming.

V. COMPUTATIONAL EXPERIMENTS

A. Experiments

We implemented the three models RNN, LSTM and BERT on our chosen Sentiment 140 dataset [6] in order to ascertain, analyze and contrast whether the result obtained from the BERT model was significantly better than the ones from RNN and LSTM.

B. Evaluation Metrics

We used Accuracy, Precision and Recall as our evaluation metrics. Accuracy is the percentage of accurate predictions. Precision is the ratio $t_p/(t_p + f_p)$ where t_p is the instances of true positives and f_p the instances of false positives. The precision is a classifier's ability to not label a sample as positive when it is negative and vice versa calculated with respect to the classification that it is being calculated against. Recall is the ratio $t_p/(t_p+f_n)$ where t_p is the instance of true positives and f_n the instance of false negatives. The recall is the classifier's ability to correctly find all the positive samples or correctly find all the negative samples based on what it is calculated relative to [25].

C. Implementation Details

We ran our experiments on Google Colab using the programming language Python. For BERT we made use of the GPU provided by Colab to parallelly process the data. For running the models RNN and LSTM, we used Google's library Tensorflow [28], NLTK for all the preprocessing [27], pretrained BERT model to run BERT [5], Scikit-Learn [29] and NumPy [30] for the statistical analysis and Plotly [31] for the plotting.

D. Experimental Results

The experimental accuracy for our dataset is summarized in the Tables 1, 2 and 3.

Table 1: ACCURACY FOR DATASET [6]

Model	Experimental Accuracy
RNN	70.2%
LSTM	71.1%
BERT	78.0%

It is quite apparent from Table 1 that BERT performed considerably better than RNN and LSTM. The accuracy could have been even higher had there been longer strings but twitter tweets were limited to 140 characters at the time that the dataset was formulated. Therefore, the benefit of BERT in capturing long term dependencies are not illustrated very well. With that being said, the performance of BERT could be said to be acceptable.

Table 2: PRECISION FOR DATASET [6]

Labels	RNN	LSTM	BERT	Instances
0	0.73	0.78	0.76	1012
1	0.69	0.67	0.79	988

It is seen from Table 2 that for the label 0, LSTM has the highest precision at 0.78 with BERT being a close second at 0.76. For label 1, BERT performs considerably better than the other two at 0.79.

Table 3: RECALL FOR DATASET [6]

Labels	RNN	LSTM	BERT	Instances
0	0.67	0.60	0.81	1012
1	0.74	0.83	0.74	988

From table 3, it can be said that BERT outperformed both RNN and LSTM quite significantly for the class 0 but LSTM had better recall for the class 1.

E. Discussions

According to the established thesis we managed to illustrate BERT outperforming both RNN and LSTM and provided computationally superior and accurate results which goes on to explain a lot of the drawback of RNN and LSTM that BERT improves upon.

VI. CONCLUSION

A. Summary

This paper looked to analyze the comparative performance of BERT with traditional models RNN and LSTM on Sentiment Analysis on the Sentiment 140 dataset. We used the same preprocessing for the 3 models to maintain consistency across the environment so that the experimental results be comparable. We successfully simulated the performance metrics which we originally set out to test – BERT outperforming both conventional sequential models RNN and LSTM on text data.

B. Future Research

For future works, we intend to look at different domains other than the one that we explored, such as sentiment analysis on product reviews, and sentiment analysis from social media posts in order to predict the mental health state of a person. We also want to test the model out on multi-class configurations with multiple labels.

C. Open Problems

BERT performs considerably better than RNN and LSTM as illustrated by this paper but despite its good accuracy, it is still plagued by some problems that are yet to be solved or addressed. BERT is a huge model so it requires a lot of storage and takes a pretty considerable amount of computational capacity to run. Another issue with BERT is dealing with class imbalances and the problems regarding scaling up the model in a way to working with larger datasets.

VII. REFERENCES

[1] G. Vinodhini and R. M. Chandrasekaran, "Sentiment analysis and opinion mining: a survey." *International Journal 2.6*, pp. 282-292, 2012.

[2] Larry R. Medsker, and L. C. Jain, "Recurrent neural networks." *Design and Applications*, pp. 64-67 p.2, 2001.

[3] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[4] Vaswani, A., "Attention Is All You Need", *arXiv e-prints*, 2017.

[5] J. Devlin, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", arXiv e-prints, 2018.

[6] Go, A., Bhayani, R., & Huang, L., "Sentiment classification using distant supervision" *CS224N project report*, Stanford, 1(12), 2009.

[7] Radford, A., Jeffrey Wu, R. Child, David Luan, Dario Amodei and Ilya Sutskever. "Language Models are Unsupervised Multitask Learners.", 2019.

[8] Liu, Yiheng, et al. "Summary of chatgpt/gpt-4 research and perspective towards the future of large language models." *arXiv preprint arXiv:2304.01852*, 2023.

[9] Fabio Duarte, "Number of ChatGPT Users (2023)", *Exploding Topics,* 16-May-2023. [Online]. Available: https://explodingtopics.com/blog/chatgpt-users#:~:text=users%20are%20American-,How%20Many%20ChatGPT%20Users%20Are%20There%3F,February%202023%20to%20April%202023). [Accessed: 28-Jul-2023].

[10] Vanaja, Satuluri, and Meena Belwal. "Aspect-level sentiment analysis on e-commerce data." *International Conference on Inventive Research in Computing Applications (ICIRCA),* IEEE, 2018.

[11] Wang, Xinyu, et al. "A depression detection model based on sentiment analysis in micro-blog social network." *Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2013 International Workshops: DMApps, DANTH, QIMIE, BDM, CDA, CloudSD, Gold Coast, QLD, Australia,* April 14-17, 2013, Revised Selected Papers 17. Springer Berlin Heidelberg, 2013.

[12] Howard, J. and Ruder, S., "Universal Language Model Fine-tuning for Text Classification", *arXiv e-prints*, 2018.

[13] Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catan- zaro, B., "Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism", *arXiv e-prints*, 2019.

[14] Liu, Yiheng, et al. "Summary of chatgpt/gpt-4 research and perspective towards the future of large language models." *arXiv preprint arXiv:2304.01852*, 2023.

[15] C. Rosset, "Turing-NLG: A 17-billion-parameter language model by Microsoft," *Microsoft Research*, 13-Feb-2020. [Online]. Available: https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17- billion-parameter-language-model-by-microsoft/. [Accessed: 29-Jul-2023].

[16] Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. "Aspect-Based Sentiment Analysis using BERT", *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pp 187–196, 2019.

[17] L.Zhang, S.Wang, and B.Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[18] D. Zimbra, A. Abbasi, D. Zeng, and H. Chen, "The state-of-the-art in twitter sentiment analysis: a review and benchmark evaluation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 9, no. 2, p. 5, 2018.

[19] R. DiPietro, Gregory D. Hager, 'Handbook of Medical Image Computing and Computer Assisted Intervention". Ch 21, pp. 503-519, 2020.

[20] Nair, V., & Hinton, G. E., "Rectified linear units improve restricted boltzmann machines", *Proceedings of the 27th international conference on machine learning (ICML-10)*, (pp. 807-814), 2010.

[21] Javaid Nabi, "Recurrent Neural Networks (RNNs)", *Towards Data Science*, 11-Jul-2019 [Online]. Available: https://towardsdatascience.com/recurrent-

neural-networks-rnns-3f06d7653a85 [Accessed: 29-Jul-2023].

[22] A. Chugh, "Deep Learning | Introduction to Long Short Term Memory", *GeeksForGeeks*, 31-May-2023. [Online] Available: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/. [Accessed: 28-Jul-2023]

[23] A. Mittal, "Understanding RNN & LSTM", *Medium*, 12-Oct-2019. [Online]. Available: https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e. [Accessed: 29-Jul-2023]

[24] Sherstinsky, A. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network" *Physica D: Nonlinear Phenomena, 404*, p.132306, 2020.

[25] Vaswani, A., "Attention Is All You Need", *arXiv e-prints*, 2017.

[26] P. Joshi, "Transfer Learning NLP: Fine Tune Bert For Text Classification," *Analytics Vidhya,* 26-Jul-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/07/transfer-learning-for-nlp-fine-tuning-bert-for-text-classification/. [Accessed: 29-Jul-2023].

[27] Bird, Steven, Edward Loper and Ewan Klein, Natural Language Processing with Python, O'Reilly Media Inc, 2009.

[28] Abadi, Martin et al., "Tensorflow: A system for large-scale machine learning", *In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016

[29] L. Buitinck et al. "API design for machine learning software: experiences from the scikit-learn project," *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[30] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586- 020-2649- 2

[31] Inc., P.T., "Collaborative data science", 2015. [Available: https://plot.ly.]