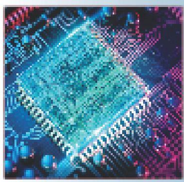
Kashif et al. LGURJCSIT 2023

LGU Research Journal of
Computer Science & IT

SSN: 2521-0122 (Online)
ISSN: 2519-7991 (Print)

doi: 10.54692/lgurjcsit.2023.0702455
Vol. 7 Issue 2, April – June 2023

Classifying Tweets with Keras and TensorFlow using RNN (Bi-LSTM)

Muhammad Kashif [1*], Dr. Ghulam Ali Mallah [2], Samar Abbas Mangi [3]
[1,2,3] Department of Computer Science, Shah Abdul Latif University Khairpur, Ghotki Campus, Pakistan

Email: muhammad.kashif@salu.edu.pk

ABSTRACT:

Due to the altering user behavior, the volume of data published by users on numerous social networking sites, including Instagram, Twitter, Snapchat, etc., has significantly increased. As a result, an enormous volume of textual, audio, and video posts—reflecting millions of users' ideas and opinions—are produced daily. Even if some posts are brief and don't matter, they provide insightful personal viewpoints on various subjects. This article employs Twitter as the analytic platform, where posts are known as tweets, focusing on identifying the underlying emotions behind such posts. This study investigates techniques for preparing and extracting Twitter data using Python-TensorFlow. Additionally, this study incorporates Recurrent Neural Network with Bi-Long Short-Term Memory (RNN (Bi-LSTM)) into the code, improving the model's performance. This research aims to effectively extract sentiment from tweets by training and evaluating the data against a classifier.

KEYWORDS: Tweet Classification, TensorFlow, Bi-LSTM, Social Networking, RNN

1. INTRODUCTION

According to [1], millions use social networking sites to express their emotions and share details about their daily activities. However, people write on various subjects, such as social relationships and product evaluations. Users can get to know and influence others by using the interactive forums that online communities offer. Additionally, social media allows businesses to engage with their customers by giving them a platform. For example, businesses can use social media to promote their products or communicate directly with customers to solicit feedback on their products and services. On the other hand, customers are completely in charge of what they want to see and when they react. This has the effect of making the company's success and failure public and causing word-of-mouth advertising.

However, the social network can influence consumer behaviour and judgment. For instance, [1] reports that 87% of web users consider customer evaluations before deciding or purchasing. Therefore, it would be more advantageous for businesses to set up to react quickly and develop a strong strategy to compete with their rivals if they can do so; more quickly. [2], [3] describe the problem. Companies and other data users still struggle with data extraction, although software can extract information about a person's thoughts about a specific good or service. Slang and colloquialisms are used in informal language, which makes use of verb tenses such as "would not" and "wouldn't" [4]. The inability of some algorithms to distinguish sentiment from the use of informal language may hinder analysis and decision-making. Emoticons, visual representations of human facial emotions [5], help a receiver understand the tone or mood of a speaker's nominal, verbal message and can improve or alter how that communication is interpreted [6]. For instance, the symbol for happiness is ☺. The current systems lack the data necessary to deduce emotions from emoticons.

Humans frequently use emoticons to express ideas that are difficult to put into words [6]. The organization is at a loss if it cannot analyze this. Even with short text services, the abbreviated form is frequently employed. On Twitter, short-form will be utilized more frequently to assist, reduce the number of characters used. This is due to Twitter's character length restriction, 1 4 0 [7]. For starters, "Tba" stands for; 'To be announced'.

2. RELATED WORK

In recent years, significant research has been done on applying machine learning methods to text classification tasks, such as sentiment analysis and tweet classification. The difficulty of categorizing tweets and comprehending their context has been addressed using various strategies. This research examines some pertinent works in the field in this section.

RNNs have been investigated in earlier research as a text classification tool. Researchers presented convolutional neural network (CNN) architecture.[8] is for phrase categorization, showcasing how well it can capture local contextual data. Long Short-Term Memory (LSTM) networks have been frequently used due to their capacity to capture long-range dependencies in sequential data, even though CNNs have demonstrated promising outcomes. A variation of RNN called LSTM, which [9] developed, has been effectively used in several natural language processing (NLP) tasks. [10] Presented deep-learning framework using LSTM for sentiment analysis of tweets in the specific context of tweet classification. Through their work, researchers showed how LSTM can effectively capture both past and future context in tweet sequences, which helped to improve sentiment classification accuracy.[11], who concentrated on multi-label tweet classification using RNNs, is another noteworthy scientific endeavour. To identify the semantic connections between tweets and their associated labels, they used LSTM and attention processes. The attention method improved the model's capacity to concentrate on specific tweets during the classification process on pertinent elements of the tweet.

Although the works described above have significantly improved tweet classification using RNNs, they have mainly concentrated on sentiment analysis or multi-label classification.This research adds to the body of knowledge by utilizing a Bi-LSTM architecture with Keras and TensorFlow. Sorting tweets into a few predetermined categories is our primary goal. By doing this, researchers expand upon the methodology and results of earlier similar studies.

3. Long-Short-Term Memory (LSTM)

The LSTM's shown efficacy in sentiment analysis supports the employment of neural networks specifically in this investigation. Although RNNs have excelled at several jobs, it is well recognized that they have a severe flaw; the vanishing-gradient issue [12] makes it challenging for the network to form long-distance connections. This problem is partially fixed by the network's use of memory units. These memory units decide what the network should remember and what it should forget. These components are present in LSTM networks. Cell state is the most significant LSTM concept. The network can keep or lose track of information using the cell state as its memory [13]. The cell status refers to how information is transferred throughout the network. This information is added or subtracted by three gates, the input, output, and forget gates, each defined by Equations (1) - (3). These gates employ a sigmoid activation function, denoted by the symbol g [14], [15]. The following is a list of the equations for the gates. Uppercase variables signify matrices, while lowercase variables signify vectors. The weights of the recurrent connections are contained in the matrices Uq and Wq. The input, q index may be either the input gate I, the output gate (o), or the forget gate (f): • The following definitions apply to the input gate, which decides what new data will be kept in a cell [16]:

$$i_t = g(W_i\, x_t + u_i\, h_t\_1 + b_i) \qquad (1)$$

The final output of the LSTM block is provided to the activation function at timestamp t through the output gate, which is defined as:

$$O_t = g\ (W_0 x_t + u_0 h_{t-1} + b_0\ ) \qquad (2)$$

The following is a description of the forget gate, which
determines which data should be forgotten:

$$f_t = g\ (W_o x_t + U_0 h_{t-1} + b_0) \qquad (3)$$

The initial values are c0 = 0 and h0 = 0. The variables are:

1) $x \in R^d$: represents a vector with dimension d for the LSTM unit.

2) $f_t \in (0,1)^h$: activation vector of the forget gates.

3) $i \in (0,1)^h$: activation vector of input gates.

In Figure 1 below, this method is depicted. Long-term dependence on phrases can be learned because the cell can retain information for extended periods without interfering with gradient descent. This is the primary argument in favour of LSTM networks. Backpropagation methods combined with a supervised environment can be used to train this network.

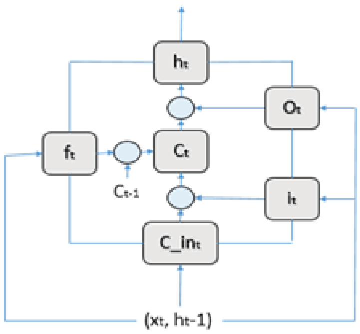

Figure 1: LSTM's cell gates and mechanism

4. LSTM and Bi-LSTM

An LSTM and a Bi-LSTM neural network implementation are used as a prediction model. These models use the vector created in the preceding stage as their input. Given that they have shown helpful in recognizing sentiment, these networks are regarded as the most successful models for sentiment prediction. Naturally, the processes for making a test set are the same as those previously mentioned. The eight distinct similarity measures are contained in the classifier's input vectors. The LSTM and BiLSTM predictors are trained using identical training data using the 10-fold cross-validation method. 90% of the tweets in each fold are utilized for testing, while 10% are used for training. No new information was used, and no additional statistical calculations were performed.

The model can make predictions just on the test set after training. The training set's rules were used to construct the test set, which can be used to evaluate the model's effectiveness.

5. Methodology

This research used a methodology in this investigation that included data collection, preprocessing, feature extraction, model architecture design, model training, model evaluation, comparison with baselines, consideration of computational resources, and guaranteeing repeatability of the experimental setting. The team collected data using a publicly accessible dataset of tweets that included a wide range of themes and emotions. The dataset comprised labelled tweets that we attempted to categorize into preset classes or categories. Before training the models, we performed data preprocessing operations such as tokenization, stop word removal, and handling of special characters or URLs. Word embeddings were used to extract information from the tweets that could be fed into the neural network. Specifically, pre-trained word embeddings were used, which capture the semantic links between words, like Word2Vec or GloVe. To improve the tweet content's depiction RNN with LSTM units were chosen for tweet classification as the model architecture.The contextual dependencies in the tweet sequences were intended to be captured by the LSTM layers. The LSTM layers were designed with a particular number of unmade proper activation function choices to reduce overfitting and used dropout regularization. For the model to perform even better, attention mechanisms and L2 regularization has been used.

This study used the Keras and TensorFlow frameworks for model training. For the objective of classifying tweets, an adequate loss function and optimization techniques were constructed. The learning rate, batch size, and number of training epochs were among the fine-tuned hyperparameters. We used regularization strategies such as early stopping or learning rate scheduling to avoid overfitting during training. The dataset was divided into training, validation, and test sets to achieve an unbiased assessment. The held-out test set was used to test the model, and the results were presented.

Our model was assessed, and its performance contrasted with other baseline models or conventional machine learning classifiers. Careful selection of the baselines and accompanying justifications revealed the superior performance and effectiveness of the RNN-LSTM model for tweet classification. Regarding computational resources, a high-performance computer cluster outfitted with GPUs was used for the tests to speed up the training procedure. Specific software versions of the Keras and TensorFlow libraries were used to maintain consistency and reproducibility.

5.1. Twitter Sentiment Corpus Dataset:

A '1' denotes a positive attitude, and a '0' a negative sentiment for each row of the 1,578,627 recognized tweets in the Twitter Sentiment Analysis Dataset. It is suggested to use 10% of the corpus for algorithm testing and 90% for sentiment classification algorithm training. The results of classifying this dataset using a straightforward Naive Bayesian algorithm were 75% accurate. Given that a guesswork approach will eventually achieve an accuracy of 50%, a straightforward approach could essentially produce results that are 50% better than guesswork. This is not ideal, but since 10% of sentiment classification by living beings is typically debatable (especially regarding communication and social sentiment classification), machine learning will likely achieve the highest accuracy possible.

Naturally, enhancing the plan with natural language processing can provide more context and emphasize textual elements more likely to lead to sentiment deduction. Analyzing this dataset using the Python-based NLTK (Natural Language Tool Kit) was enjoyable. NLTK offers a versatile and customizable framework for diverse natural language analysis and classification methodologies, making it a valuable tool for this research study.

One thing to remember is that tweets, or any social casual communication, frequently employ truncated letters, characters inside of words, overused punctuation, and might not follow standard grammar rules; one can either normalize this usage when classifying text or use it. Consider how many exclamation marks are used in a communication to determine its intensity. Rather than a mild (or neutral) mood, this could be a sign of powerful positive or negative emotion.

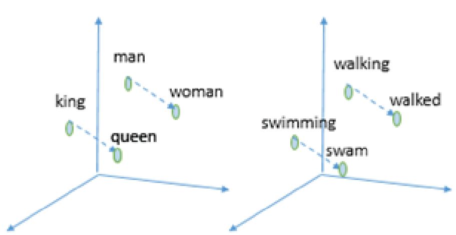

Figure 2: Examples of vector

The research becomes increasingly exciting when breaking down the sentiment of a statement (or a tweet in this case) in connection to various elements (or nouns) within that phrase.Figure 2shows the examples of vector embedding that show how words relate to one another analogously.

5.2. Setting up our dtaset:

This research uses ThinkNook-displayed Twitter Sentiment Analysis Dataset.

Opening the dataset, shows a big sheet with the following rows and columns:

Table 1: Showing the first few rows and columns of the dataset.

ItemID	Sentiment	Sentiment Source	Sentiment Text
1	0	Sentiment 140	Is so sad for my APL friend….
2	0	Sentiment 140	I missed the New Moon Trailer…
3	1	Sentiment 140	Omg it's already 7:30☺…

Only columns 1 and 3 hold significance, with the neural network being trained using inputs from the "Sentiment Text" column and outputs from the "Sentiment" column.

5.3. Performance Evaluation

The implementation utilized Python 3's Keras module and TensorFlow. Colab Notebooks can be optimized for massive arrays using Google GPUs, a free cloud-based alternative to Jupyter Notebooks. These laptops feature a 12GB NVIDIA Tesla K80 GPU in their hardware, capable of running nonstop for up to 12 hours. Empirically, the following optimization hyperparameters were selected: Batch size is 500; the Keras Sequential model's Embedding layer; the SpatialDropout1D layer, which removes 1D feature mappings to promote feature independence; and the LSTM and Bi-LSTM layer are the other features. The input pattern can only be 280 characters, the maximum length for a tweet. The model comparison has been done with the following article [17].

Table 2 shows the comparison of our model (RNN+Bi-LSTM) with other relevant models. Our proposed model defeated the relevant models by a reasonable margin.

Table 2: Results and comparison with other models

Model	Average accuracy (%)
Unigram	56.58
Senti-features	56.31
Kernal	60.60
Unigram + Senti-features	60.50
Kernal+ senti-features	60.83
Unigram + Bigram	78.88
RNN+Bi-LSTM (our)	82.3

6. CONCLUSION

This research paper's conclusion described the methodology for categorizing tweets using Keras and TensorFlow with an RNN + Bi-LSTM architecture. The methodology used word embeddings and other properties to improve tweet representation. The RNN + Bi-LSTM model correctly classified tweets by properly capturing contextual dependencies.

The model performed better than benchmark models, demonstrating its sentiment analysis, subject detection, and opinion-mining capability. Future research can investigate more complex RNN architectures and various pre-trained embedding and tackle noisy and skewed Twitter data issues.

The created model demonstrates its capacity for comprehending and analyzing social media data, providing new opportunities to understand better the large volume of data posted on websites like Twitter.

REFERENCES

[1] M.Rambocas; and J. Gama, "Marketing research: The role of sentiment analysis." Marketing_research_The_role_of_sentiment_analysis (accessed Aug. 01, 2023).

[2] R. R. V. Kadam, "Sentiment Analysis using Tweets." https://grdjournals.com/article?paper_id=GRDJEV04I050018 (accessed Aug. 01, 2023).

[3] Y. Wang, J. Guo, C. Yuan, and B. Li, "Sentiment Analysis of Twitter Data," Appl. Sci. 2022, Vol. 12, Page 11775, vol. 12, no. 22, p. 11775, doi: 10.3390/APP122211775, Nov. 2022.

[4] Y. Zhou and Y. Fan, "A Sociolinguistic Study of American Slang," Theory Pract. Lang. Stud., vol. 3, no. 12, pp. 2209–2213, doi: 10.4304/TPLS.3.12.2209-2213, 2013.

[5] T. Anuprathibha and C. S. K. Selvib, "A survey of twitter sentiment analysis," IIOAB J., vol. 7, no. 9 Special issue, pp. 374–378, doi:10.26438/IJCSE/V6I11.644648, Aug. 2016.

[6] P. Kavitha and M. Prabakaran, "An Efficient Tweeter Sentiment Analysis Sfcetr Selective Feature Based Case Content Extraction Using Maximum Entropy Classifier To Rank The Tweets," Int. J. Comput. Sci. Eng., vol. 6, no. 9, pp. 289–299, doi: 10.26438/IJCSE/V6I9.289299, Sep. 2018.

[7] D. Boyd, S. Golder, and G. Lotan, "Tweet, tweet, retweet: Conversational aspects of retweeting on twitter," Proc. Annu. Hawaii Int. Conf. Syst. Sci., doi: 10.1109/HICSS.2010.412, 2010.

[8] Y. Kim, "Convolutional Neural Networks for Sentence Classification," EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf., pp. 1746–1751, doi: 10.3115/V1/D14-1181, , 2014.

[9] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, "A survey on long short-term memory networks for time series prediction," Procedia CIRP, vol. 99, pp. 650–655, doi: 10.1016/J.PROCIR.2021.03.088, Jan. 2021.

[10] X. Zhang, J. Zhao, and Y. Lecun, "Character-level Convolutional Networks for Text Classification," Adv. Neural Inf. Process. Syst., vol. 2015-January, pp. 649–657, Sep. 2015, Accessed: Aug. 02, 2023. [Online]. Available: https://arxiv.org/abs/1509.01626v3, Sep. 2015, Accessed: Aug. 02, 2023.

[11] J. Read and F. Perez-Cruz, "Deep Learning for Multi-label Classification," [Online]. Available: https://arxiv.org/abs/1502.05988v1,- Dec. 2014, Accessed: Aug. 02, 2023.

[12] M. Roodschild, J. Gotay Sardiñas, and A. Will, "A new approach for the vanishing gradient problem on sigmoid activation," Prog. Artif. Intell., vol. 9, no. 4, pp. 351–360, doi: 10.1007/S13748-020-00218-Y, Dec. 2020.

[13] F. Huang, X. Li, C. Yuan, S. Zhang, J. Zhang, and S. Qiao, "Attention-Emotion-Enhanced Convolutional LSTM for Sentiment Analysis," IEEE Trans. neural networks Learn—Syst., vol. 33, no. 9, doi: 10.1109/TNNLS.2021.3056664,2022.

[14] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative Study of CNN and RNN for Natural Language Processing," [Online]. Available: http://arxiv.org/abs/1702.01923,Feb. 2017, Accessed: Aug. 01, 2023.

[15] Y. Ito, "Approximation of continuous functions on Rd by linear combinations of shifted rotations of a sigmoid function with and without scaling," Neural Networks, vol. 5, no. 1, pp. 105–115, doi: 10.1016/S0893-6080(05)80009-7, 1992.

[16] F. A. Lovera, Y. C. Cardinale, and M. N. Homsi, "Sentiment Analysis in Twitter Based on Knowledge Graph and Deep Learning Classification," Electron. 2021, Vol. 10, Page 2739, vol. 10, no. 22, p. 2739, doi: 10.3390/ELECTRONICS10222739,, Nov. 2021.

[17] Neha, H. Gupta, S. Pande, A. Khamparia, V. Bhagat, and N. Karale, "Twitter Sentiment Analysis Using Deep Learning," IOP Conf. Ser. Mater. Sci. Eng., vol. 1022, no. 1, p. 012114, doi: 10.1088/1757-899X/1022/1/012114, , Jan. 2021.