

An Optimized Deep Learning Model for Emotion Classification in Tweets

Chinu Singla¹, Fahd N. Al-Wesabi^{2,3}, Yash Singh Pathania¹, Badria Sulaiman Alfurhood⁴,
Anwer Mustafa Hilal^{5,*}, Mohammed Rizwanullah⁵, Manar Ahmed Hamza⁵ and Mohammad Mahzari⁶

¹Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India

²Department of Computer Science, King Khalid University, Muhayel Aseer, Kingdom of Saudi Arabia

³Faculty of Computer and IT, Sana'a University, Sana'a, Yemen

⁴Department of Computer Science, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Saudi Arabia

⁵Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, Alkharj, Saudi Arabia

⁶Department of English, College of Science & Humanities, Prince Sattam bin Abdulaziz University, Alkharj, Saudi Arabia

* Corresponding Author: Anwer Mustafa Hilal. Email: a.hilal@psau.edu.sa

Received: 26 May 2021; Accepted: 13 August 2021

Abstract: The task of automatically analyzing sentiments from a tweet has more use now than ever due to the spectrum of emotions expressed from national leaders to the average man. Analyzing this data can be critical for any organization. Sentiments are often expressed with different intensity and topics which can provide great insight into how something affects society. Sentiment analysis in Twitter mitigates the various issues of analyzing the tweets in terms of views expressed and several approaches have already been proposed for sentiment analysis in twitter. Resources used for analyzing tweet emotions are also briefly presented in literature survey section. In this paper, hybrid combination of different model's LSTM-CNN have been proposed where LSTM is Long Short Term Memory and CNN represents Convolutional Neural Network. Furthermore, the main contribution of our work is to compare various deep learning and machine learning models and categorization based on the techniques used. The main drawback of LSTM is that it's a time-consuming process whereas CNN do not express content information in an accurate way, thus our proposed hybrid technique improves the precision rate and helps in achieving better results. Initial step of our mentioned technique is to preprocess the data in order to remove stop words and unnecessary data to improve the efficiency in terms of time and accuracy also it shows optimal results when it is compared with predefined approaches.

Keywords: Meta level features; lexical mistakes; sentiment analysis; count vector; natural language processing; deep learning; machine learning; naive bayes



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Sentiments are something that one can express through various ways as it could be verbal, written or over the internet. Natural Language Processing and Python provide a well-developed tool with which one can easily get rid of the lexical adulterations and focus on the actual context to successfully predict the sentiments for the same.

Twitter is the world's largest micro-blogging site and it has become ubiquitous these days across the world. It allows for short posts primarily containing text that is 140 characters long and these are referred to as "tweets" [1]. Twitter in particular among all social media platforms has a widespread adoption and a rapid communication volume. Twitter has 313 million users active within a given month and 100 million users actively tweeting. It has a wide range of applications in commerce, public health [2,3], opinion detection about political tendencies [4,5] and stock market monitoring [6]. Emotion analysis is the process of identifying the attitude towards a target or topic. The attitude can be the polarity (positive or negative) or an emotional state such as joy, anger, or sadness [7]. Thus, manual classification of posts and opinion mining becomes an inflexible option. The very subjectivity of the data has made this problem is still an open problem in the field.

Natural Language Processing is concrete with tradition text genres like news data and long summaries of books and papers. But twitter possess an entirely different challenge for natural language processing, they are short and have "hash tags #" (which are a type of tagging for Twitter messages). The language used is also highly informal with creative spelling, slang, new words and URLs and gender-specific abbreviations like RT for "Re-Tweet." This paper aims to overcome these hurdles in the pre-processing phase to boost the results of the different models.

For this study, a proper benchmark of the different models is being set and the dataset incorporated is SemEval Datasets [8] which is an ongoing semantic evaluation of computer semantics and has widely been used for benchmarking sentiment analyses. In our research paper, CNN as well as LSTM is being used as both of them individually comprises of certain shortcomings which can be mitigated by using both of them in tandem and this is done by using a hybrid model as mentioned in this paper .CNN has been used to create a pool layer that is further passed to the LSTM model down the pipeline. The main contributions of our approach, hybrid LSTM and CNN can be summarized as follows:

- Majority of the work done in this field aims over a large data to extract any essence regarding any particular trend that is being carried out by using machine learning model to avoid the overhead issue, the same is used in our model despite it being a neural network.
- Most of the work done focuses on word embedding by extracting important words from tweets which fails to extract the essence of it like sarcasm and irony this can be resolved by further dividing the word embedding into regions and using a convolution layer to extract further features.
- Most of the research papers do no lay emphasis on emoticons which are essential for social media platform and hence sentimental analysis without a bias towards them is not accurate.
- Thus, this framework provides a more robust scalable and more functional approach that can be tailored fit to meet special needs like understanding the sentiments of the public about various situations more accurately and thus can help in decision making and social engineering.

Another hurdle that this paper tends to solve is the extensive use of emoji's that has drawn growing attention from research and these emoji's contain important information as previous

work has shown that, it is useful to pre-train a deep neural network on an emoji prediction task with pre-trained emoji to predicting emotion and sarcasm with greater accuracy [9]. In our work an “opinion lexicon” has been used for a lexicon-based approach [10] that uses punctuations and emoticons as key flags for determination. Previous literatures lack in consideration of this complexity and variety of emoji’s. Therefore, applying emoji embedding to older models could result in boosting their accuracy and give better overall results. Our work is to compare various deep learning and machine learning models and categorization based on the techniques used. Furthermore, in this paper, hybrid combination of different model’s LSTM-CNN have been proposed to achieve more accuracy and efficiency by reducing latency.

The remainder of the article is structured as follow. In Section 2, we present the related work already done in this field. The proposed model is explained in Section 3. Implementations and simulation is provided in Section 4. Comparison and discussion are provided in Section 5. Results analysis is provided in Section 6, and finally, we conclude the article in Section 7.

2 Related Work

Various Machine learning and Deep learning methods have been introduced for sentiment analysis of tweets. State-of-the-art systems [11,12] used approaches of integrating different models and applied feature vectors including semantics, syntactic features, and word embeddings to display tweets. There have been a lot of ways in which researchers have used to classify Reddit comments as “depressed” or “not depressed” [13]. The paper uses a BERT [14] based model, and a neural network with a word embedding (CNN) model for classification. Their results showed that CNN without embedding performed better than the BERT based model. Lexical corrections and intensive preprocessing with an Apache System (SVM) [15] have been used to formulate a model with 5% greater accuracy than the traditional sentiment analysis method. Though the research clearly mentions in the conclusion that the result will subjected to change to a large extent depending on the dataset used.

This recent study [16] primarily focused on twitter uses an unsupervised approach for a graphical representation of opinions all in real time applied over scale dataset for the years 2014–2016. Sony [17] an open source java based social dynamics analyzer with its main focus being user influence and event detection. Lambda architecture is also software architecture and is used along-side machine learning to analyze large data streams such as twitter. Lexical resources like Wilson et al. [18] labeled a list of English words in positive and negative categories. ANEW [19] application for twitter has uses the AFINN lexicon which is a list of words rated from –5 to 5 based on how positive or negative they are. Research papers as written by extensively used the Stanford NLP library which produces great results for abstraction and removal of data. Different Deep learning models have been utilized to develop end-to-end systems in many tasks including text classification, speech recognition and image classification. Results show that these types of systems automatically extract high-level features from raw data [20,21].

2.1 *Bidirectional Encoder Representations from Transformers (BERT)*

The reference to this model has been drawn from [22] led by It is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP task [23]. Fig. 1 depicts the architectural layout of BERT Model.

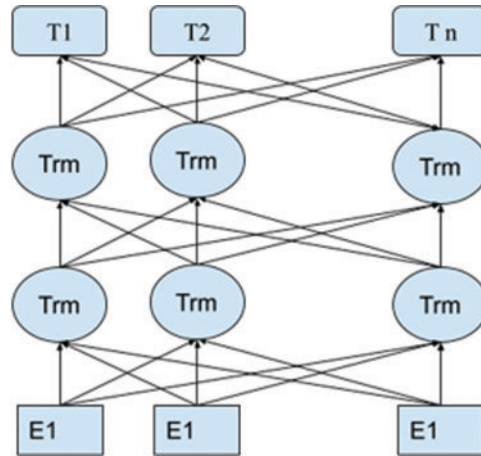


Figure 1: Architecture of BERT model

It is pre-trained on a large corpus of unlabeled text including the entire Wikipedia and Book Corpus this has made Bert a well know NLP model in all tasks including twitter sentiment analysis. The BERT model implemented is Bert-tokenizer for classification of text using Tensor Flow 2.0 the batch size used is 30, dropout rate of 0, 2 and 10 epochs and accuracy of 87% has been obtained by the model.

2.2 Convolution Neural Networks (CNN)

The architecture for this convolution network is shown in Fig. 2 [24]. The input in the model is treated in the form of a sequence of words and for each of these words a sentence matrix is built in which each column represents a word embedding at the corresponding position.

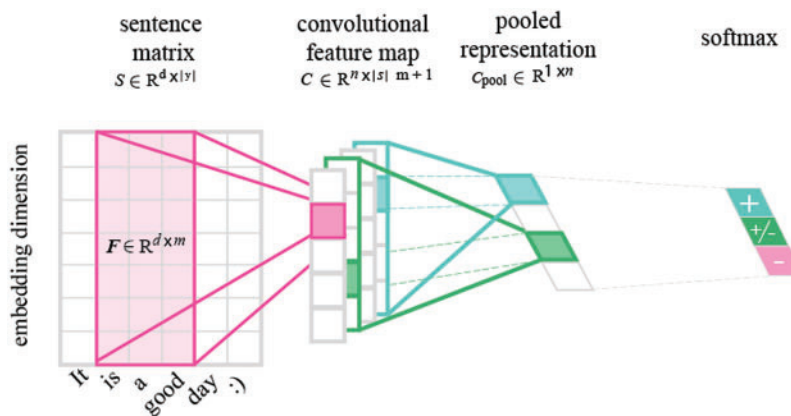


Figure 2: Architecture of CNN

Due to CNN being heavily dependent on computation power it takes the most amount of time and resources. In CNN model, the layers which are present closer to the input in ConvNet help in classifying the basic rudimentary features for classification such as postv lexical indicators like “good,” “bad” these are then taken into account and then the next layer performs a more

detailed evaluation thus the top layers in the end compiles all complex features and makes a prediction, the same can be observed in Fig. 2.

2.3 Long Short-Term Memory (LSTM)

LSTM is a artificial Recurrent Neural Network (RNN) is used in the field of deep learning to perform sentiment analysis due to its nature to grasp embeddings long term dependencies in discrete text groups.

The first step is to embed words to word as shown in Fig. 3 followed by step 2 where the RNN receives a sequence of vectors as input and consider the order of the vectors to generate sentiment as compared to a basic LSTM model usually comprised of a single hidden LSTM layer and then a feed forward output layer. This implies we will have fixed length reviews encoded as integers and then be converted to embedded vectors pushed to a LSTM in a recursive fashion and pick the last prediction as an output sentiment.

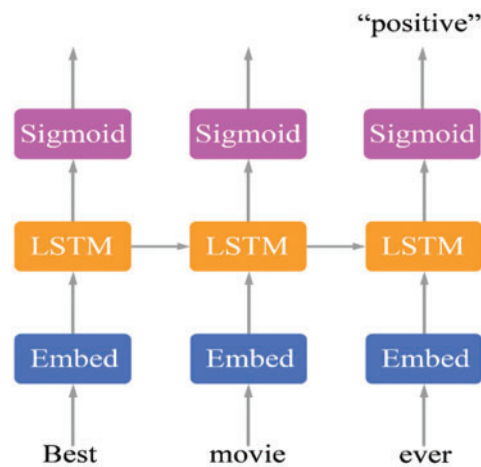


Figure 3: LSTM architecture

It was observed that when the words were encoded to integers randomly the accuracy was about 60–65% but when the encoded happened based upon the frequency of the words the accuracy jumped up to 83–87%. The words to vector transformation were carried out by using word2vec library.

3 System Model

After analyzing different research papers described in Section 2, it is clear that deep learning-based text classification using NLP is a popular topic nowadays. Identifying positive and negative emotions from social media is a very popular and challenging topic for research. As many people share their emotions through social media, it can be easy to collect huge data for research. In this paper, we have proposed some models to classify emotions from tweets of Twitter users. For this purpose, data preprocessing is an essential step after collecting the dataset.

The main contribution of the proposed model can be summarized as follows:

- Comparing various deep learning models and traditional machine learning models. This has been done to make sure this paper serves as the basis for further research into fine tuning various methods to a more custom fit for a particular task.
- To preprocess a data with various lexicon corrections that range from general cleaning of tweets to fixing slang used to a certain degree. As Twitter being a micro-blogging site it's filled with noise and various nonstandard ways of writing text, our robust preprocessing system aims to correct this to provide with clean noise free data. This ensures better results even with models that are highly sensitive to noise like SVN.
- Propose a hybrid model using LSTM and CNN to improve both accuracy of sentiment analysis and the time taken over huge datasets. That has been greatly customized with an initial layer and a final layer. The initial layer comprises of pre trained word embedding from "Google News Word 2 VCC" with a word vector vocabulary of more than 3 million words.
- In our LSTM + CNN technique we use CNN to create the initial classification layers that our then fed to the LSTM model which greatly reduces time and increases accuracy it can serve to be a great way to quickly analyze twitter sentiment without taking up too much time and has proven to be better than many standard models such as BERT.

This research focus on proposing both baseline models (traditional machine learning models) and deep learning models (Stacked LSTM, Stacked LSTM with 1D convolution, CNN with pre-trained word embedding, a BERT based model). For baseline models, TF-IDF (term frequency-inverse document frequency) and count vectors features have been used for Multinomial Naive Bayes, Support Vector Machine (SVM) and Logistic Regression separately as input. TF-IDF is a weighting scheme that assigns each term in a document a weight based on its term frequency (TF) and inverse document frequency (IDF). The terms with higher weight scores are considered to be more important [25]. Count Vector works on Terms Frequency, i.e., counting the occurrences of tokens and building a sparse matrix of documents x tokens [26]. For all deep learning models, epoch size has remained the same and it is 3 because of the large size of the dataset. But batch size, dropout probability, activation and optimization has been varied from model to model. The main target of this proposed method is to obtain the best model for sentiment analysis that can then easily be fine-tuned to match particular topics or genres as shown in Fig. 4. This has been carried out in this paper by complete testing of different machine learning models and deep learning models. This was done by using sent 140 dataset that was preprocessed for noise and the various adulteration found in micro blogging sites (slang, re-tweets etc). We also propose a hybrid model that uses LSTM in combination to computer neural networks to develop a flexible yet fast model that can be adapted to any situation. This was done by adding a one-dimensional CNN and a max pooling layer after the embedding layer which is the feeds the consolidated features to the LSTM this is explained further in great detail in Section 4.4.4.

4 Implementation

4.1 Conformance

Consider Intel R core i7-7700 CPU is the CPU used for this research work. The computer has 16 GB RAM and a Windows operating system paired with a 1050ti GPU ere. Thus, the language used for the results obtained is python while the idea being used to run the codes is Visual Studio code. All the models mentioned in Section 4 have been tested on the same hardware for uniform results to compensate for the technological advancements in the past few years.

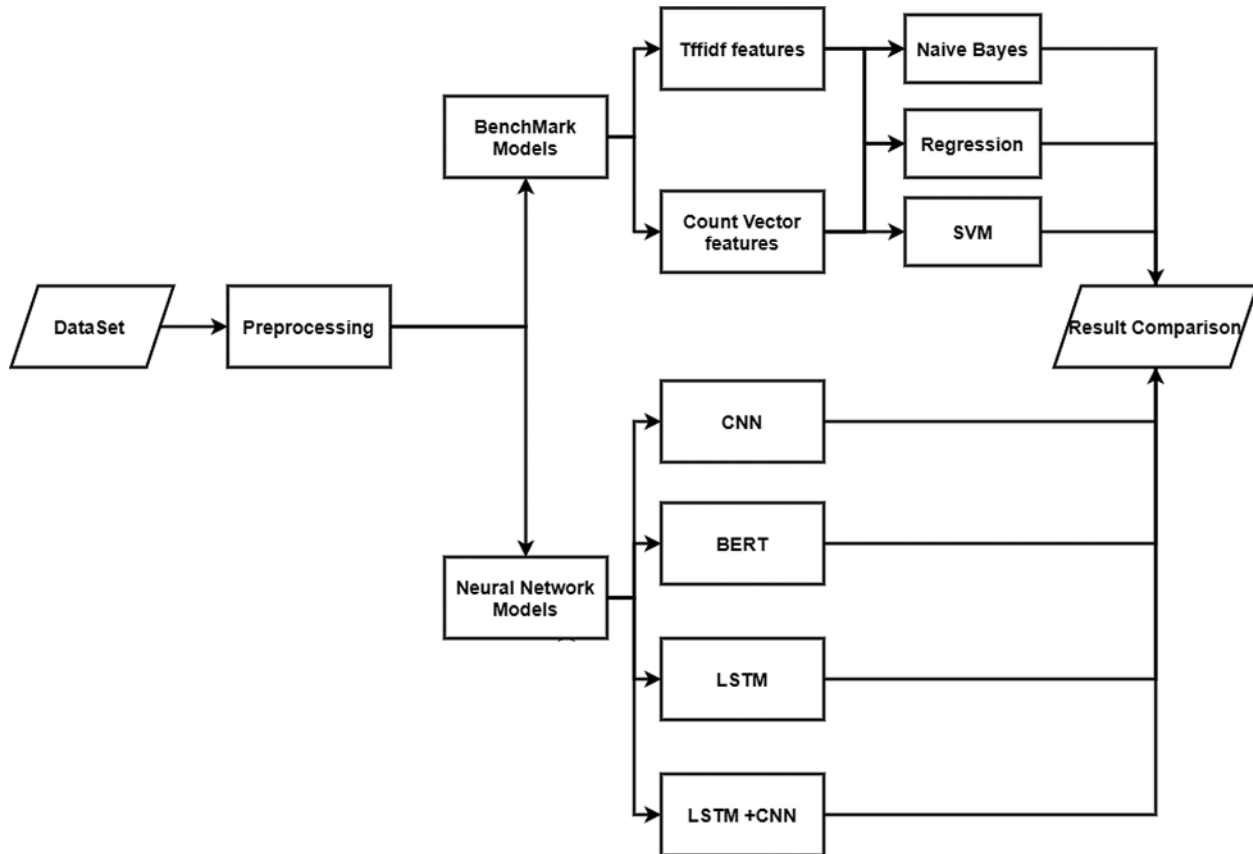


Figure 4: Overview of proposed work

4.2 Standard Dataset

The training and testing data has been extracted from sentiment140 [27] which comprises 1048576 rows with 6 columns. But for the requirement of this study, only two columns have been extracted which constitutes of the follow:

- The polarity of the Tweet
- The text of the tweet

For the first column aforementioned the data is binary 0 pointing to negative data while 4 points to a positive emotion tweet this can be seen in Fig. 5 along with the units in percentage.

This dataset was extracted using twitters own Application programming interface and thus contains of real world tweets and not prefabricated hand chosen tweets that could skew the result in any way. The dominant topics of which the tweets are extracted are shown in Tab. 1.

The dataset has been repeatedly used for research purposes for sentiments analysis because of it consisting of opinions from a spectrum of topics leading to no such bias in the dataset.

The dataset consists of 6 columns but only two columns the text and the sentiment were required for the scope of this research and thus have been used.

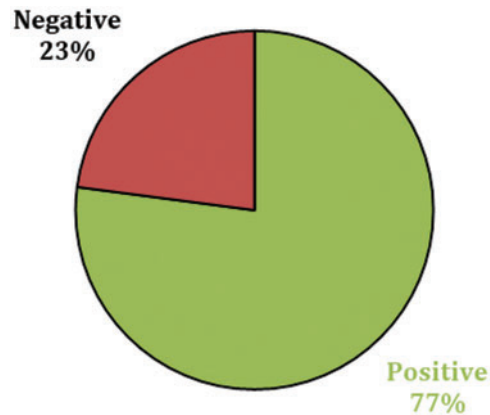


Figure 5: Classification of dataset

Table 1: Primary topics in Dataset

Sr. No.	Topic	Sr. no.	Topic
1	Product	8	Crypto currency
2	Company	9	Sports
3	Music	10	Politics
4	Misc.	11	Science
5	Event	12	Religion
6	Movie	13	Video games
7	Location	14	Miscellaneous

4.3 Preprocessing

The next step after collection of dataset is to preprocess it so that it can be removed of all the garbage values and unnecessary data that might have been left out due to this being a real life dataset. This helps in filtering out some of the values from the huge dataset. Since twitter is a micro-blogging website where people often use “hashtags”, “URL”, ”slangs”, “acronyms” and other lexical complexities it becomes very important to regulate it this was done by implementing these steps broadly.

- Tokenizing words.
- Removing hashtags and URLs.
- Changing uppercase to lowercase.
- Reserved words (RT, FAV1).
- Removing unnecessary and multiple pronunciations.
- Removing stop words.

Most of this was achieved through a package called tweet-preprocessor 0.60 [28] which is an inbuilt library widely used to clean tweet of the know noise present in them in the form of above-mentioned occurrences.

The next step in preprocessing was normalizing text for lexicons was done [29] by changing words with repeated syllables to more understandable structures as depicted in Fig. 6 but the latter is easily recognizable in different lexicons than the prior and hence this normalization is necessary.



Figure 6: Example showing lexical correction

4.4 Generic Models

There have been three generic models that have been used to act as a base like to compare to the deep learning models mentioned above, these models serve as a yard stick to analyze which deep learning models performs how much better the generic models used were as follows.

4.4.1 Naïve Bayes

Baye's rule as shown in Eq. (1) points to the explanation of an event on prior knowledge [30] the probability of event. A (positive) given that B (negative) has already occurred after Laplace smoothing naïve Bayes results in a mathematical Eq. (2).

$$\frac{P(\text{pos})}{P(\text{neg})} \prod_{i=1}^m \frac{P(w_i | \text{pos})}{P(w_i | \text{neg})} \quad (1)$$

$$p\left(\frac{x'}{\text{positive}}\right) = (\text{number of review with } x' \text{ and target_outcome} = \text{positive}) / (N = a * k) \quad (2)$$

where a: smoothing parameters, K: no of features, N: number of reviews with target positive

Here the product of these ratios can result in too large numbers these are then resolved by taking logs on either side or comparing log of prior to log of likelihood this was carried out for 1/4 of the allocated test split of the dataset to get baseline accuracy. For the creation of this model Python's scikit-learn's Multinomial has been used.

4.4.2 Support Vector Model (SVM)

The SVM model thus implemented is a linear one-vs.-rest Support-Vector-Machine (SVM) classifier [31]. That takes into account each unique word present in the sentence, as well as all consecutive words. For the representation in the dataset to be useful all of which was converted into a vector, where the vector is the same length as our vocabulary, i.e., the list of all words observed in our training data, with each word representing an entry vector. If a particular word is present, that entry in the vector is 1, otherwise 0. This was done using the count vector present in sklearn.

The model was made using python's scikit-learn's SGDC classifier with the attributes of maxiter = 12 and alpha = 0.001.

4.4.3 Regression Models

As a baseline even the logistic regression model from Python's scikit-learn has been used the sigmoid function used in logistic regression as shown in Eq. (3).

$$h(x^i, \theta) = \frac{1}{1 + e^{-\theta^T * x^i}} \quad (3)$$

where, θ : weight parameter, and x^i : some input parameter.

4.4.4 LSTM and CNN

A stacked residual LSTM model inspired from [24] also used as adding more layers to LSTM accounting to 8-layer neural network has seen better performance than lexicon base or conventional NN-based techniques [24] hence a Word2seq CNN + LSTM model is implemented with 1 embedding layer 2 convolution layer with max pooling 1 LSTM layer and 2 fully connected layers. Ten epochs are used, and the total runtime was 80 mins.

This model using model checkpoints and call-backs, saves the model weights when validation accuracy is maximum thus the pooling layer can use the standard length of 2 to halve the feature map size. Fig. 7. The CNN model initially benefits from pre trained word embeddings which were used the pre trained word embeddings were obtained from Google News Word2Vec which includes a word vector for a vocabulary for 3 million words and phrases that are trained over 100 billion words. The training and validation accuracy and validation loss with these pre-trained word embeddings for CNN has shown in CNN with word embeddings are used to extract complex features from sentences while LSTM is used as a classifier.

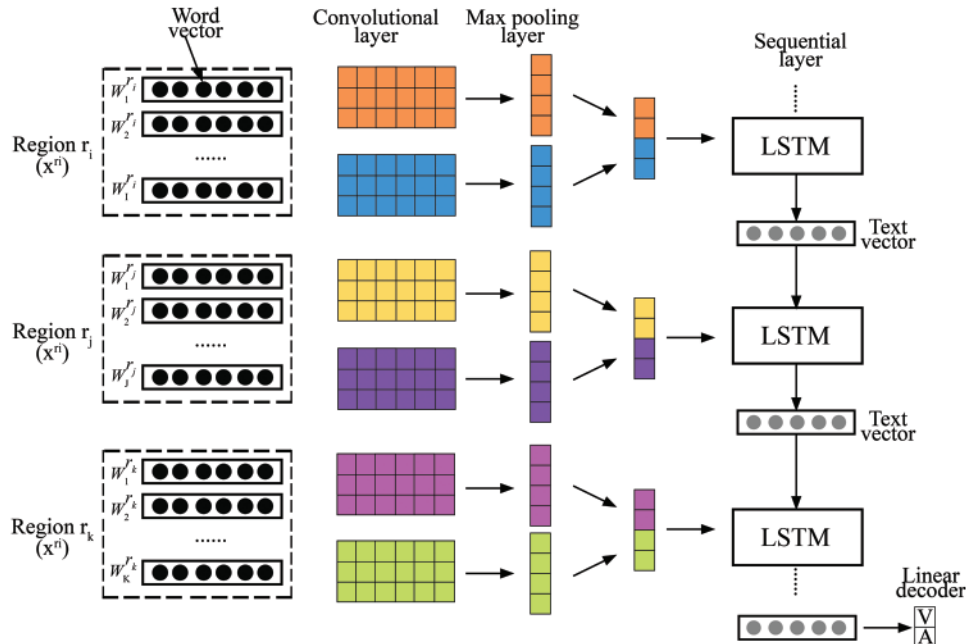


Figure 7: Overview of LSTM- CNN hybrid

The hybrid structure shown in Fig 7 incorporates a CNN network over regions r that denotes subsets of the datasets to find structure 2 embedding layers which is then incorporated into 1

pooling layer that increases the speed of the LSTM model. This is done after embedding the word vectors which was seen to have a severe impact on the CNN training time shown in Fig 8.

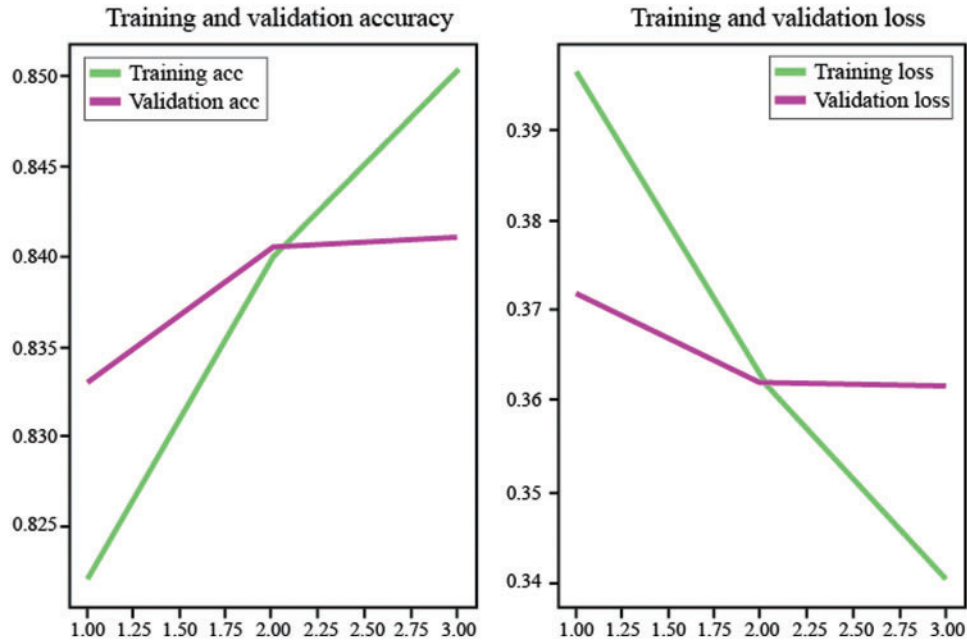


Figure 8: Training and Validation accuracy and validation loss for CNN with word embeddings

5 Comparison and Discussion

In this section, comparison of models used will be explained. It can be clearly observed from Figs. 9 and 10 that among all the rudimentary baseline models, SVM with TF-IDF features outperforms the rest of the rudimentary features.

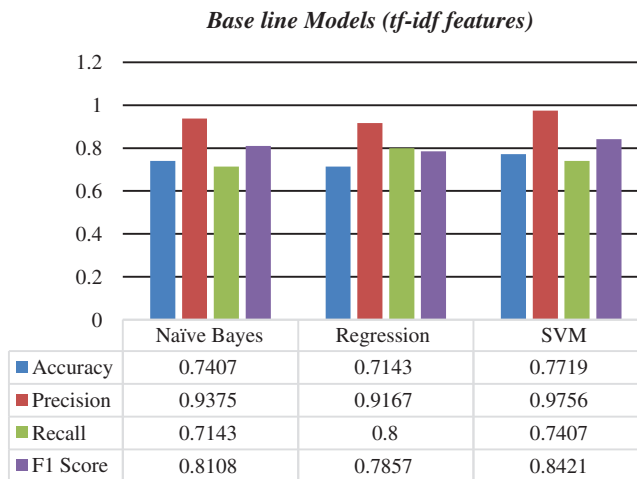


Figure 9: Resulting benchmarks (TF-IDF)

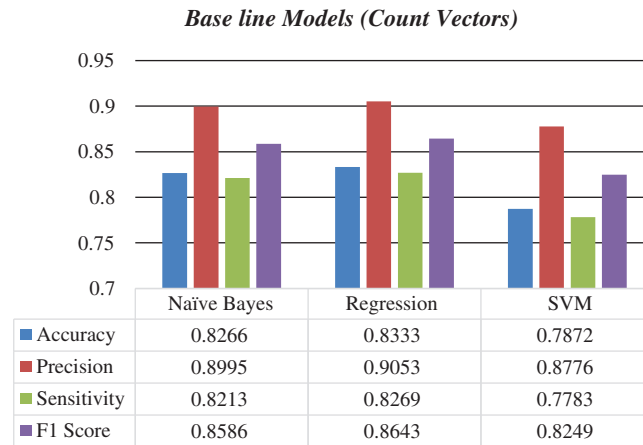


Figure 10: Resulting benchmarks (count vectors)

While SVM remains constant it takes a huge hit during the initial stages the pre-processed data is used without lexicon corrections and other twitter specific cleaning strategies as mentioned in Section 4.3, this could primarily pointed towards the variations noise in SVM model. These are corrected but it does not show much variation between count vector and TF-IDF features in comparison to other models.

Fig. 11 shows the comparison of different models when count vectors were used to TF-IDF features on the basis of accuracy as follow.

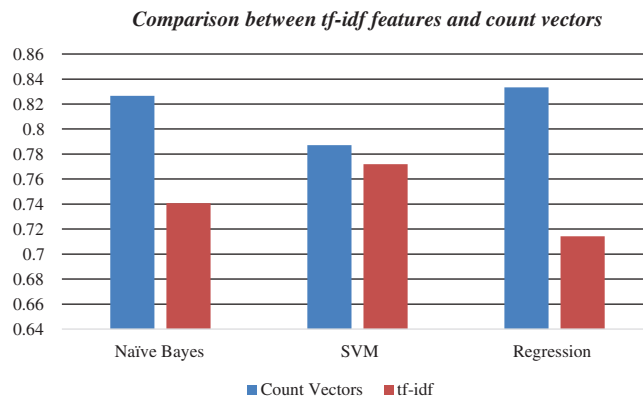


Figure 11: Comparison of different models

The various benchmarking parameters for different deep learning models have been shown in Fig. 12. The results obtained are judged on the parameters of accuracy, precision, sensitivity and F1 score it has been clearly seen that all the deep learning models performed very well across the board the high sensitivity across the models fell in line with the fact that all the models are trained on dataset which had more positive value than negative values runtime is not taken into consideration while comparing the models but LSTM and CNN performed much faster due to maximum polling in comparison to all other deep learning models. Simulation results depict that

deep learning models outperform and show significant improvements than traditional machine learning approaches.

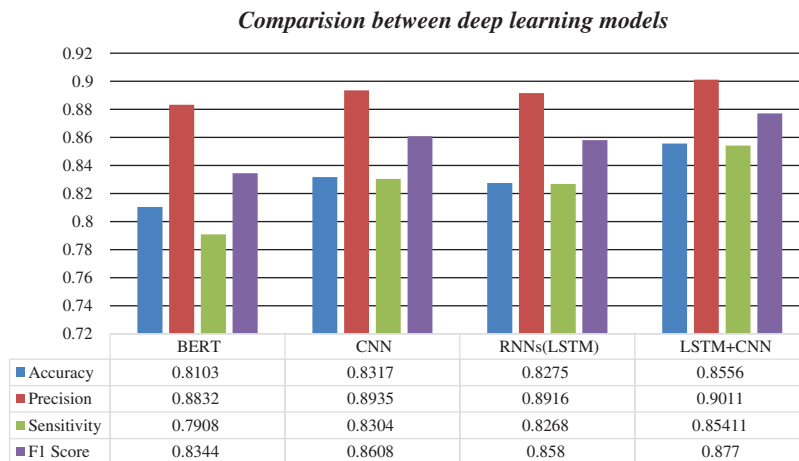


Figure 12: Resulting benchmarks (Deep Learning Models)

6 Result Analysis

The result accounted for after formulating all mentioned methods give us much greater insight of how these works and which are better for sentiment analysis since our dataset has only positive and negative emotions and the theoretical chance was around 50% but all the models especially the deep learning models have performed far better than the theoretical expectations.

Almost all of the deep learning models performed better than the traditional machine learning models signifying how important it is for a model to be adaptive when it comes to Natural Language Processing. In baseline models, logistic regression has performed the best with 83% accuracy followed by Naïve Bayes and SVM on further inspection this can be attributed to the fact that the inconsistency of the lingo on social media websites can create a lot of noise to which SVM is very sensitive [32].

In deep learning models, the best performing model is LSTM with a combination of CNN with 0.877 F1-Score and 90% Precision.

Twitter users use a lot of emoticons to express how they feel which have been considered with these models as such. In future, the model can also be scaled to accommodate for bilingual dialects which have not been well documented or worked up for us to scale the project to at this moment.

7 Conclusion

This paper is presented to serve as baseline comparison as to which model work better when it comes to modern day language processing problems that require modern day solutions it is clear that with proper word preprocessing and embedding deep learning models tend to perform very well across the board having greater than 80% accuracy which can then be specialized for the particular trend to get even better results.

No particular stop words or specialized texts are used for this research both of which can increase accuracy when building a model with a particular even in focus. The models used are working in a very rudimentary form with little to no specialization which can be done but, in our case, would have skewed the results.

The project demonstrates a working functional LSTM + CNN hybrid model that can be scaled up quickly and perform better overall than existing solutions. This can be incorporated into previous research papers [33] that utilize machine learning models over neural networks due to the overhead that comes with it.

It is seen that lexicon correction and preprocessing play a major role in determining sentiments when it comes to non-uniform stream of textual data. This was seen to have heavy impact on SVM models and thus correcting noise becomes essential.

The hybrid solution thus provided in this research paper helps in fast deployment and extensive scaling due to the CNN tokenization leveraging the speed of a CNN network while having high accuracy benefiting from an LSTM model towards the end of the pipeline.

Funding Statement: The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work under Grant Number (RGP. 2/23/42), www.kku.edu.sa. This research was funded by the Deanship of Scientific Research at Princess Nourah bint Abdulrahman University through the Fast-Track Path of Research Funding Program.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] M. Fiona, D. Carin and H. Gail, "Understanding twitter," *British Journal of Occupational Therapy*, vol. 76, no. 6, pp. 295–298, 2013.
- [2] Y. Chen, Y. Zhou, S. Zhu and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *Proc. of the 2012 Int. Conf. on Privacy, Security, Risk and Trust (PASSAT), and 2012 Int. Conf. on Social Computing (SocialCom)*, Amsterdam, The Netherlands, pp. 71–80, 2012.
- [3] C. Cherry, S. M. Mohammad and B. Bruijn, "Binary classifiers and latent sequence models for emotion detection in suicide notes," *Biomedical Informatics Insights*, vol. 5, pp. 147–154, 2012.
- [4] S. M. Mohammad, X. Zhu, S. Kiritchenko and J. Martin, "Sentiment, emotion, purpose, and style in electoral tweets," *Information Processing Management*, vol. 51, pp. 480–499, 2015.
- [5] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intelligent Systems*, vol. 31, pp. 102–107, 2016.
- [6] H. Y. Yun, P. H. Lin and R. Lin, "Emotional product design and perceived brand emotion," *International Journal of Advances in Psychology Research*, vol. 3, pp. 59–66, 2014.
- [7] S. M. Mohammad, "Sentiment analysis: Detecting valence, emotions, and other affectual states from text," in *Emotion Measurement*, Meiselman, H. L., Ed.; Wood head Publishing: Cambridge, UK, pp. 201–237, 2016.
- [8] N. Iqbal, A. M. Chowdhury and T. Ahsan, "Enhancing the performance of sentiment analysis by using different feature combinations," in *Int. Conf. on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, University of Rajshahi, Bangladesh, pp. 1–4, 2018.
- [9] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in *Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing, EMNLP*, Copenhagen, Denmark, pp. 1615–1625, 2017.

- [10] X. Ding, B. Liu and P. Yu, "A holistic lexicon-based approach to opinion mining," in *Proc. of the 2008 Int. Conf. on Web Search and Data Mining WSDM*, New York, NY, United States, pp. 231–240, 2008.
- [11] P. Goel, D. Kulshreshtha, P. Jain and K. Shukla, "Prayas at emoint 2017: An ensemble of deep neural architectures for emotion intensity prediction in tweets," in *Proc. of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Copenhagen, Denmark, pp. 58–65, 2017.
- [12] V. Duppada, R. Jain and S. Hiray, "Seernet at semeval-2018 task 1: Domain adaptation for affect in tweets," in *Proc. of the 12th Int. Workshop on Semantic Evaluation, The Association for Computational Linguistics*, Stroudsburg, vol. 2, pp. 18–23, 2018.
- [13] M. M. Tadesse, H. Lin, B. Xu and L. Yang, "Detection of depression-related posts in reddit social media forum," *IEEE Access*, vol. 7, pp. 44883–44893, 2019.
- [14] J. Devlin, M. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *NAACL-HLT 2019*, Minneapolis, Minnesota, vol. 1, pp. 4171–4186, 2019.
- [15] F. Bravo, M. Mendoza and B. Poblete, "Combining strengths, emotions and polarities for boosting twitter sentiment analysis," in *Proc. of the Second Int. Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM'13)*, Association for Computing Machinery, New York, NY, USA, pp. 1–9, 2013.
- [16] N. Azzouza, K. Astouati, A. Oussalah and S. AitBachir, "A real-time twitter sentiment analysis using an unsupervised method," in *Proc. of the 7th Int. Conf. on Web Intelligence, Mining and Semantics (WIMS'17)*, Association for Computing Machinery, New York, NY, USA, pp. 1–10, 2017.
- [17] A. Guille, C. Favre, H. Hacid and D. Abdelkader, "An open-source platform for social dynamics mining and analysis," in *Proc. of SONDY*, United State, pp. 1005–1008, 2013.
- [18] M. Wilson, W. Moss, E. Helen and V. Halen, "Perceptual distance and competition in lexical access," *Journal of Experimental Psychology, Human Perception and Performance*, vol. 22, pp. 1376–1392, 1997.
- [19] F. Årup, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," in *Proc. of the ESWC2011 Workshop on 'Making Sense of Microposts': Big Things Come in Small Packages 718 in CEUR Workshop Proc.*, Ithaca, USA, pp. 93–98, 2011.
- [20] Y. Cun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.
- [21] D. Tang, B. Qin and T. Liu, "Deep learning for sentiment analysis: Successful approaches and future challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, pp. 292–303, 2015.
- [22] M. Pota, M. Ventura, R. Catelli and M. Esposito, "An effective BERT-based pipeline for twitter sentiment analysis: A case study in Italian," *Sensors*, vol. 21, no. 1, pp. 1–21, 2021.
- [23] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," in *Proc. of the 38th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'15)*, Association for Computing Machinery, New York, NY, USA, pp. 959–962, 2015.
- [24] J. Wang, B. Peng and X. Zhang, "Using a stacked residual LSTM model for sentiment intensity prediction," *Neurocomputing*, vol. 322, pp. 93–101, 2018.
- [25] S. Kim and J. Gil, "Research paper classification systems based on TF-IDF and LDA schemes," *Human-centric Computing and Information Sciences*, vol. 9, no. 30, pp. 1–21, 2019.
- [26] R. Leuret and R. Collobert, "Rehabilitation of count-based models for word vector representations," in *Int. Conf. on Intelligent Text Processing and Computational Linguistics*, Konya, Turkey, Springer, Cham, vol. 9041, 2015.
- [27] K. Grandmaster, Sentiment analysis with tweets, 2017. <https://www.kaggle.com/kazanov/sentiment140>.
- [28] Tweet-preprocessor. 2020. [Online]. Available: <https://pypi.org/project/tweet-preprocessor/>.
- [29] V. Ngoc, C. Shivade, R. Ramnath and J. Ramanathan, "Towards building large-scale distributed systems for twitter sentiment analysis," in *Proc. of the 27th Annual ACM Symp. on Applied Computing (SAC'12)*, Association for Computing Machinery, New York, NY, USA, pp. 459–464, 2012.
- [30] P. Kaviani and S. Dhotre, "Short survey on naive Bayes algorithm," *International Journal of Advance Research in Computer Science and Management*, vol. 4, no. 11, pp. 607–611, 2017.

- [31] S. Naz, A. Sharan and N. Malik, "Sentiment classification on twitter data using support vector machine," in *IEEE/WIC/ACM Int. Conf. on Web Intelligence (WI)*, Melbourne, Australia, pp. 676–679, 2018.
- [32] A. Feizollah, S. Ainin, N. B. Anuar, N. A. Abdullah and M. Hazim, "Halal products on twitter: Data extraction and sentiment analysis using stack of deep learning algorithms," *IEEE Access*, vol. 7, pp. 83354–83362, 2019.
- [33] S. Albahli, A. Algsham, S. Aeraj, M. Alsaeed, M. Alrashed *et al.*, "Covid-19 public sentiment insights: A text mining approach to the gulf countries," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 1613–1627, 2021.