Performance Analysis of Neural Networks forOffensive-Hinglish Detection

1st Bhushan khot
Department of Information Technology
Pillai College Of Engineering
New Panvel, India

2nd Kartik Chauhan
Department of Information Technology
Pillai College Of Engineering
New Panvel, India

3rd Aryan Agrawal
Department of Information Technology
Pillai College Of Engineering
New Panvel, India

4th Naqi Khatib
Department of Information Technology
Pillai College Of Engineering
New Panvel, India

5th Prof. Varunakshi Bhojane
Department of Information Technology
Pillai College Of Engineering
New Panvel, India

***Abstract*—The proliferation of smartphones as it becomes more affordable, is giving many a means to express their hateful and offensive ideas unrestrictedly. Social media websites are the main outlet of such expressions which have inadequate systems for hate speech detection especially for less common languages. Hinglish being a pronunciation based pseudo language, offensive message detection is even more difficult. In the paper, possible techniques to counter this problem have been discussed with main focus on different combinations of text vectorization and neural networks. Count and TF-IDF Vectorizers have been crossed with different neural networks trained on a corpus of Hinglish texts acquired from the Twitter API 2.0. LSTM, BiLSTM, RNN based models were trained and tested. The results consist of a comparative study of these neural networks showcasing RNN model to be working with Count Vectorization most efficiently.**

***Index Terms*—Hinglish, Hate, Vectorization, Count, TF-IDF, LSTM, BiLSTM, RNN**

I. Introduction

The fleeting growth of social media on users has led to them spending a significant amount of time on many online messaging and content markets apps to share information, and to explore their interests. Online hate speech is defined as written messages that include derogatory or discriminatory words. Companies are more concerned than ever before about hate speech content because the majority of it now-a-days occurs in the form of code-mixed languages such as Hinglish. Throughout the COVID-19 pandemic lockdown, children and young teenagers spent increasingly more time on the internet leading to a 70% increase in online hate speech instances that involved minors. While impressionable minds benefit from the outreach of online social influence by learning from educational content and motivational speakers, they at the same time are at a risk of cyber-bullying or bad influence. Identifying offensive content online has become a pressing need in safeguarding adolescent online safety, as they are more vulnerable to the detrimental effects of biased and harmful material.

Over the last 10 years, India has seen some incredible improvements to technologies in handheld devices, also being a progressively developing nation, there are a lot of young minds to be influenced. India's population is projected to cross over 1.4 Billion in 2024, the amount of data generated by Indians alone is tremendous. To our use case, this data comes in the form of hate speech on the internet. Hindi being the more popular language, most of it is typed in a code-mixed language - Hinglish which is Hindi words in Roman text. This new pseudo-language has no basic grammar and very less rules to be governed with.

In this paper, data acquisition and augmentation along with the mix of vectorization techniques with model building is analyzed and evaluated. A total of 6 different models are trained on a Hinglish text dataset, with the main objective being the performance analysis of different embedding techniques with different neural networks.

II. Related Work

A. Offensive Language and Hate Speech Detection with Deep Learning and Transfer Learning [1]

The study in this paper suggests a method for categorizing tweets into three categories: hateful, offensive, and neither of those. In studies, BI-LSTM models are constructed from empty embedding using the public Twitter data set, and the same neural network is also trained using Glove Embedding. Using Pre-trained models such as BERT, and GPT-2 a transfer learning strategy for hate speech detection is shown. The best model (BI-LSTM) is subjected to a hyperparameter tuning study that considers several neural network structures, learning

rates, and normalizing procedures.When evaluated using test data, the optimal set of parameters results in an accuracy of over 92%.

B. Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data [2]

The study in this paper presents a comparison of different deep learning algorithms. Algorithms with different arrangements are tested for their efficiency. The evaluation's findings were marginally poorer. The comparatively poor performance of these techniques exposed the drawbacks of CNN and LSTM in this particular field. Because of CNNs' efficient dimensionality reduction procedure and LSTM networks' retention of word dependencies, it has been found that CNN and LSTM networks coupled perform better than when employed alone. Additionally, the system performs better when many CNN and LSTM networks are used. As different datasets perform differently in terms of accuracy, it is clear that the key to improving the performance of these systems is to have a suitable dataset. As a result, it seems that more resources and time needs to be invested in developing quality training sets

C. Detecting Offensive Language on Arabic Social Media using Deep Learning [3]

This paper addresses the challenge of detecting offensive language on Arabic social media. The author want to control Arabic-language social media material and shield users from the negative impacts of provocative remarks including verbal abuse and hate speech. This study addresses four distinct neural network models like LSTM, Bi-LSTM and combined CNN-LSTM., CNN+ LSTM.The results imply that the CNN works well in terms of accuracy and precision despite only being able to learn local characteristics from word n-grams.The combined CNN-LSTM network, on the other hand, outperforms the individual CNN-LSTM networks in terms of recall since it has the ability to learn long term dependencies through its LSTM layer.

D. Detecting Offensive Tweets in Hindi-English Code-Switched Language [4]

In the study, the performance of the Ternary Trans-CNN model was evaluated using macro criteria including accuracy, F1 score, precision, and recall were used instead, as the class imbalance did not significantly alter the outcomes. The model was first trained on ordinary dataset A and its performance was taken as the baseline. However, when tested on dataset HEOT without transfer learning, the model's performance decreased compared to the baseline. This is due to the degradation of the syntax of Hinglish tweets in the dataset HEOT after transliteration and translation, which results in a loss of contextual structure in the tweets. After retraining the Trans-CNN model, its performance on dataset HEOT not only significantly improved but also surpassed the earlier results on dataset A. This indicates that there was a positive transfer of features from the source to the target data.

E. Deep Learning for Hate Speech Detection in Tweets [5]

In this paper, the use of different deep neural network modes for the purpose of detecting hate speech was investigated. This paper includes study of three different algorithm each combined with different vectorization technique. This paper discovered that its approach greatly outperformed the current approaches. The best accuracy values were obtained when gradient-boosted decision trees were paired with embeddings learnt from deep neural network models.

III. DATASET

The offensive hinglish text dataset is a collection of twitter data that has been annotated for offensive language. The dataset contains a total of 24,000 messages and tweets that are labeled as either offensive or non-offensive.

Each message or tweet in the dataset is labeled as either 0 or 1, where 0 indicates non-offensive language and 1 indicates offensive language. The dataset is skewed a little towards the offensive message class as there slightly increased false positive rate is tolerable in our use-case.

The text data in the dataset includes a variety of offensive language, including profanity, hate speech, and obscene messages. The messages and tweets were collected from the social media platform, Twitter using its own latest Twitter API 2.0

The dataset is reduced to contain only what is needed and provided in a .csv format, with each row representing a single instance of text data. The columns in the dataset include the following:

Text: The text of the message or tweet

Label: 0 for non-offensive, 1 for offensive

TABLE I
SAMPLE OFFENSIVE TEXT

Tweets	Lable
Jaa na l**de apni g**d mara	Offensive
Muh band kar ch***ye	Offensive
Kaisa hai mere bhai	Not-Offensive

A. Data augmentation

There is an abundance of English datasets of offensive tweets in comparison to Hinglish ones. This fact is leveraged to our advantage in augmenting our Hinglish dataset. In order to save data collection resources and time spent, the already trusted and verified datasets were translated and transliterated to generate new data.

Popular English datasets were chosen and using Google Translate API and some pre-processing, Hinglish data was created. The procedure was to first, identify all nouns and adjectives in a sentence through spaCy library's Part-of-Speech tagger component. After which, the identified keywords were

isolated and the rest of the sentence translated to Hinglish. The isolated keywords were fixed and did not go the process. The next step was to create a dictionary containing rarely used words and their corresponding slang terms. As the process often resulted in classical Hindi words which are not used by the newer generations, this step was needed for real world data representation. The final step was to manually go through each data points and retain only the ones which were logically correct to our use-case.

Also other techniques were used to augment data like Synonym Replacement and Word Swapping but to limited extents. The overall dataset was also put under limits in accordance to our processing capabilities because of training time and infrastructure constraints.

TABLE II
SAMPLE GENERATED OFFENSIVE TEXT

Tweets	Lable
langda kardunga ma***aat ma**rc**d	Offensive
par mera way ko f**k yo kutiya mujhe jaisa year	Offensive
tu akela kaafi hai inke liye	Not-Offensive

IV. METHODOLOGY

Deep Learning models are way more effective than machine learning models in taking into consideration the interdependencies and automatic feature selection. The course of steps that was undertook involved the following steps right from data tuples to model evaluation -

A. Text Pre-processing

Pre-processing is the first step to be done after data acquisition and reduction. The data collected was scanned for special characters, numbers, other script characters, etc. Tweets containing usernames, mentions and hashtags were cleaned and all text was put in lowercase. Tokenization has been done by using the tokenizer from NLTK Library after which the text is finally ready for conversion.
A list of custom stop-words for the Hinglish language was developed considering in all some of the variations of each word. A total number of 120 stop-words.

B. Vectorization

Word Embeddings or Vectors*[6]* are a type of textual representation in NLP to map words or a group of words from known vocabulary to a corresponding mathematical representation used to convey meaning to a processor. The dataset consisting of raw texts must be converted into numeric representations in order to process it via a CPU. The two techniques that have been used are -

- Count Vectorization - The most basic type of vectorization technique that numbers all the words in a collection sequentially and assigns ”0” to Out-Of-Vocabulary words (OOVs). The value in each cell of the matrix represents the count of the corresponding word in the corresponding document. While count vectors are useful for analyzing text based on word frequency, they have several significant drawbacks.
  - They cannot differentiate between important and unimportant words for analysis.
  - They give priority to words that are more abundant in the corpus, even if they are not the most significant.
  - Count vectors do not identify relationships between words, including linguistic similarities.
- TF-IDF Vectorization - TF-IDF (Term Frequency-Inverse Document Frequency) is a word embedding generation technique used to calculate the value of a word or term in a document with respect to its value in a collection of documents. It is a mathematical formula to assign a number to each word which represents their frequence in a document in comparison with other documents. Essentially, TF-IDF quantifies the importance of a word in a document.
  TF-IDF scores a word by multiplying the word’s Term Frequency with the Inverse Document Frequency.
  - Term Frequency: It is calculated by dividing the frequency of a word in document by the sum of the number of words in the document.

$$TF = \frac{the\ number\ of\ times\ term\ appear\ in\ the\ document}{total\ number\ of\ terms\ in\ the\ document}$$

  - Inverse Document Frequency- This terms signifies the commonality of a word to appear in other documents. It is a measure of the number of times a word is found in the whole corpus of documents. For example, common words like ”aur”, ”tu”, ”bhai” are assigned lower importance.

$$IDF = \log\left(\frac{the\ number\ of\ times\ term\ appear\ in\ the\ document}{total\ number\ of\ terms\ in\ the\ document\ contain\ the\ term + 1}\right)$$

C. Model Creation

After creating meaningful word embeddings, these are required to be fed into a neural network. The shape of these word embeddings are 2-dimensional in the case of Count Vectorizer and 3-dimensional in the case of TF-IDF vectorization.
The two mentioned vectorization techniques were combined with 3 different architectures to give 6 different techniques of sentiment detection. These architectures were -

- Simple RNN *[7]* - a combination of basic dense and dropout layers

- LSTM *[8]* - The main catch of an LSTM model is the memory cell called as a 'cell state' that stores its state over the many iterations of traning. The cell state is the horizontal line like structure passing through inputs and store them if they match a certain criterion. It can be seen as a conveyor belt on which information just passes, unchanged.

- BiLSTM *[9]*- Each training sequence is fed into the recurrent unit forwards and backwards so as to take into account meanings of both ways. Both sequences feed data to a same output layer. Bidirectional LSTMs have complete information about every point in a given sequence, everything before and after it.

D. Training & Testing

For the architectures with Count vectorization, the model has been set to train on for 10 epochs with a batch size of 16 as the training complexity is slower for CV and also the loss stagnates after the 10th epoch. TF-IDF models were trained with 64 batch size and only for 5 epochs due to resources constraint. Larger batch sizes were required for these networks as the input for embedding layer is a 3-dimensional array and takes up a lot of RAM. The same was the reason for reducing the dataset in order to reduce the size of this 3-dimensional array for LSTM and BiLSTM networks.

Adam and RMSProp optimizers *[10]* were used as the loss functions which were computed using Binary Cross-Entropy*[11]*for offensive and non-offensive classes. Dropout*[12]* layers were added in between embedding layers to avoid over-fitting of the model as the data was also limited.

V. EXPERIMENTAL ANALYSIS

A. Experimental Setup

All the six architectures were executed in Google Colab environment which provides high-end GPU, RAM and processing systems. For comparitive purposes, all models were feeded equal number of data points and were relatively of a similar nature in the number of layers and tensors. The models were trained, validated and tested. The training loss curves of the two kinds of models are shown in figure 2 and figure 3.

B. Perfomance Analysis

The augmented dataset of 24,000 data points has been fed to 6 different models each having their own specific combination of layers and parametres. Each model to be compared has been kept of similar architecture for the comparison to be fair. For example, both LSTM models have the same root layers but its parameters hypertuned to give the best results. We have used several metrics to ensure that the models are thoroughly analyzed based on the concept of confusion matrix.*[13]*

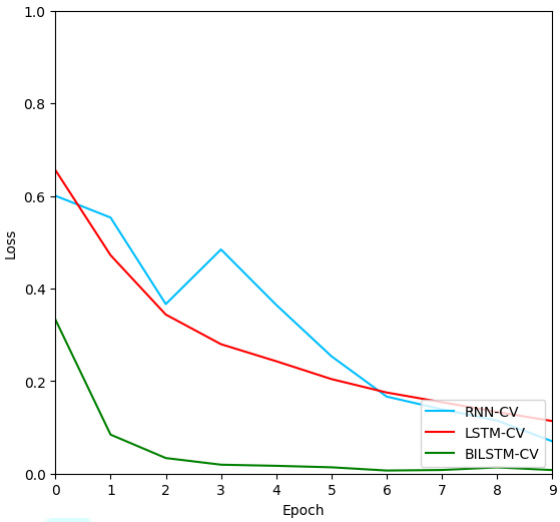

Fig. 1. CV Loss Curve.

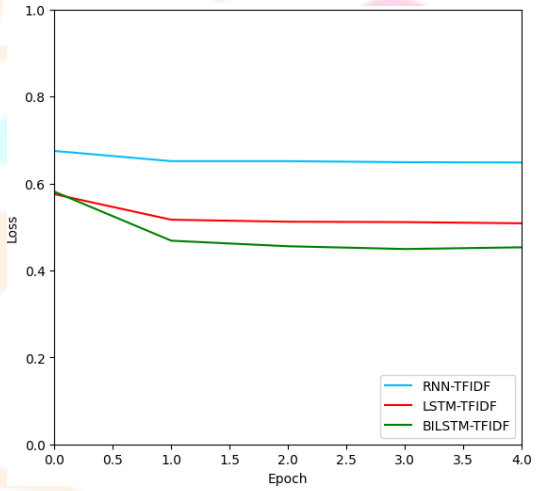

Fig. 2. TF-IDF Loss Curve.

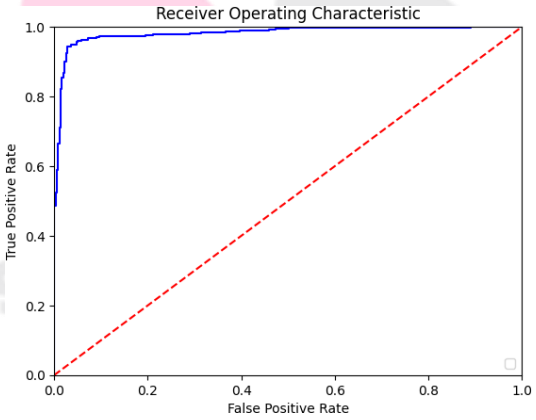

Fig. 3. ROC Curve of RNN-w-Count.

[14] ROC*[14]* curves were used to showcase the performance of the best trained classifier. As the classification is binary, The ROC curve provides a clearer picture of the outcomes.

TABLE III
COMPARITIVE ACCURACY RESULTS

	Count		TF-IDF	
Model	Training	Testing	Training	Testing
BiLSTM	99.75%	84.46%	74.86%	64.44%
LSTM	95.59%	79.98%	74.38%	73.72%
RNN	97.94%	95.33.%	57.28%	51.56%

It can be observed that, RNN with Count Vectorizer has excelled all other methods for categorising emotion due to its distinctive ability to identify patterns or catchphrases in texts.. Both BiLSTM and LSTM have outperformed the conventional approach.*[17]* used for offensive message detection on Hinglish texts. From the above stated findings, it can be concluded that RNN with Count Vectorizer is the best trained architecture on this training dataset for performing sentiment classification on Hinglish based texts.

TABLE IV
DETAILED COMPARITIVE RESULTS

Vectorization	Model	Recall	Precision	F1
Count	BiLSTM	0.90	0.9	0.91
	LSTM	0.89	0.87	0.86
	RNN	0.88	0.86	0.88
TF-IDF	BiLSTM	0.82	0.76	0.81
	LSTM	0.78	0.72	0.74
	RNN	0.8	0.71	0.76

VI. CONCLUSION

In this paper, the study of different domain technologies is presented. For training and validation data, the gathering process included Web Scraping, Official Datasets, Custom-Generated Data, etc. Different models such as Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM) and Bi-LSTM were studied and researched thorough many performance measures to find insights into analyzing and identifying offensive Hinglish language. The different hybrid approaches are also described. The comparative study of various techniques mentioned above is presented in this report. Furthermore, the prerequisite for NLP is word vectorization and it is a domain which holds a large variety of methods and techniques. Some of these have been tried and tested - TF-IDF, Bag-of-Words, GLoVe, etc. The performance measures like precision and recall are described in this report.

REFERENCES

[1] Wei, B., Li, J., Gupta, A., Umair, H., Vovor, A., Durzynski, N. (2021) "Offensive Language and Hate Speech Detection with Deep Learning and Transfer Learning" ArXiv. https://doi.org/10.48550/arXiv.2108.03305.

[2] D. Goularas and S. Kamis (2019), "Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data," International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), Istanbul, Turkey, pp. 12-17, doi: 10.1109/Deep-ML.2019.00011.

[3] H. Mohaouchane, A. Mourhir and N. S. Nikolov, (2019) "Detecting Offensive Language on Arabic Social Media Using Deep Learning," Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, 2019, pp. 466-471, doi: 10.1109/SNAMS.2019.8931839.

[4] Mathur, Puneet Shah, Rajiv Ratn Sawhney, Ramit Mahata, Debanjan (2018) "Detecting Offensive Tweets in Hindi-English Code-Switched Language" 18-26. 10.18653/v1/W18-3504.

[5] Badjatiya, P., Gupta, S., Gupta, M., Varma, V. (2017) "Deep Learning for Hate Speech Detection in Tweets". ArXiv. https://doi.org/10.1145/3041021.3054223

[6] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., Potts, C. (2011), "Learning word vectors for sentiment analysis." Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1. Association for Computational Linguistics, pp. 142-150.

[7] Alex Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network"

[8] Hochreiter, S., Schmidhuber, J., "Long short-term memory." Neural computation 9.8, 1997, pp. 1735-1780.

[9] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, "Identifying and categorizing offensive language in social media (offenseval)," arXiv preprint arXiv:1903.08983, 2019.

[10] Kingma, D. P., Ba, J. , "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980, 2014.

[11] Rubinstein, R., "The cross-entropy method for combinatorial and continuous optimization." Methodology and computing in applied probability 1.2, 1999, pp.

[12] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1, 2014, pp. 1929-1958.

[13] Sokolova, M., Lapalme, G., "A systematic analysis of performance measures for classification tasks."Information processing management 45.4, 2009, pp. 427-437.

[14] Bradley, A. P., "The use of the area under the ROC curve in the evaluation of machine learning algorithms." Pattern recognition 30.7, 1997, 1145-1159.

[15] Badjatiya, P., Gupta, S., Gupta, M., Varma, V. (2017). Deep Learning for Hate Speech Detection in Tweets. ArXiv. https://doi.org/10.1145/3041021.3054223