TOXIC COMMENT CLASSIFICATION

Ankit Sanjyal[1], Paresh Subedi[2], Rohit Kumar Jaishay[3], P. Manikandan[4]
[123] B.Tech 4th Year, CSE, Jain University, Bengaluru 562112, Karnataka, India.
[4]Associate Professor, CSE, Jain University, Bengaluru, 562112, Karnataka, India.

E-mail: [1]asanjyal56@gmail.com ; [2]pareshsubedi1@gmail.com ; [3]rjmentor2001@gmail.com; [4]p.manikandan@jainuniversity.ac.in ;

---***---

Abstract - In the recent times, the users on social media platforms have increased dramatically, leading to a significant increase in the amount of online content being generated. However, this surge in online activity has also led to a rise in toxicity and negativity on these platforms. Online communities are now inundated with toxic comments and hateful messages that can have serious psychological and emotional impacts on the targeted individuals. This has created a pressing need for effective methods to identify and remove toxic comments from online platforms to ensure a safe and inclusive environment forall users. To deal with this problem, machine learning models have been created to automatically classify comments as toxic ornon-toxic. In this research paper, we present an approach for toxic comment classification using a Bi-directional LSTM neural network architecture. We test our approach using a publicly available dataset provided by a Kaggle competition. Our experiments show that LSTM performs excellent on text classification tasks with 98% AUC score.

Keyword's: Machine learning, Neural Network,

1.INTRODUCTION

Toxic comments have become a major issue on online platforms, including social media, blogs, and news websites.Such comments can cause harm to individuals and communities, and can negatively impact online discourse. Due to the scale of the problem, it is not feasible for human moderators to manually review every comment posted on these platforms. Thus, machine learning models have been developed to automatically classify comments as toxic or non-toxic which can help in having valuable discussionsonline and can also help governments make regulations.

In this study, we suggest a Bi-directional LSTM model for toxic comment classification. Recurrent neural network of this type are very effective at processing sequential data, including text and every component of an input sequence has information on both past and future data. Also, Bi-directionalLSTM model is beneficial in some NLP tasks, such as sentence classification, translation, and entity recognition

2. LITERATURE REVIEW

The existing system of the Toxic Comment Classification emphasizes upon the basic LSTM and CNN methods to bring down the two levels of granularity, the first one being the word level and another being the character level which workson both binary and multi label classification tasks. Convolutional Neural Network (CNN) is a level of DeepLearning algorithm which works well on parts such as takinginput images and finding patterns in it, it works in such a way to recognize objects, classes and even categories. CNN depends upon the network architecture for learning from the data.

Recurrent neural networks can learn order dependency inprediction problems using Long Short Term Memory (LSTM) networks, a complicated subfield of Deep Learning. Such learning is beneficial for numerous complicated fields, including speech recognition, machine translation, and manyothers.

3.PROPOSED METHODOLOGY

In this research we propose a study of Bi-directional LSTM model on multi-label toxic comment classification, to determine whether the comment belongs to a toxic or non- toxic label. The main steps of our project are listed and described in the subsection.

a. Data acquisition

We are utilizing the Kaggle Competition dataset for this project. About 160000 labelled comments from Wikipedia talk pages make up the dataset, and those comments are classified into 6 labels of toxicity.

- toxic
- severe_toxic
- obscene
- threat
- insult
- identity_hate

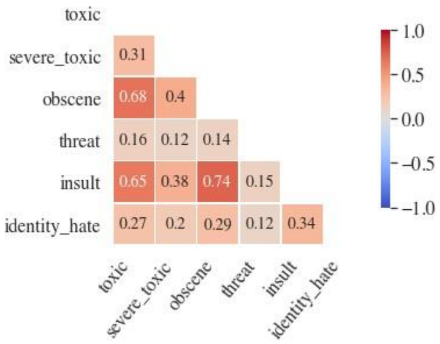


Figure [1]: Correlation Table

Our data is organized accordingly and the levels of the toxicity is defined by the level of above given points. Hence, these data are essential for training our model which can provide accurate prediction. And in the given figure correlation between the toxicity labels are depicted.

b. *Data Preprocessing*

The process of converting unprocessed data into a format better suited for analysis is known as data preprocessing. It is an essential phase in machine learning since the accuracy of the output is closely related to the quality of the input data. Data preprocessing can involve a variety including data cleaning, data transformation, and data normalization. Some common techniques used in data preprocessing include:

Data Cleaning- In this part of the project we performed different data cleaning operations such as Lower casing, removing unwanted characters, removing characters from both left and right, removing punctuations and numbers and single character removing. Regardless of how advanced our machine learning algorithm is, we cannot obtain better results from bad data. Which is why data cleaning is a very essential part of machine learning.

Tokenization- Tokenization is the process of breaking down streams of textual data into smaller meaningful elements called tokens. For example, a sentence can be a token of a paragraph, a word can be a token of a sentence. This turns an unstructured textual data into a numerical data structure that is suitable for machine learning.

In our project we performed tokenization by first creating an object instance of tokenizer class with argument num_word equal to 100000 which is nothing but the number of words to work with. Then we will be passing the dataset as a parameter in “fit_on_texts” then the list of words are converted into tokens using “tokenizer texts_to_sequences”.

Word Embedding- Word embedding is a technique where each word is represented as vectors that captures inter-word semantics. Words that are close in meaning will have similarity in the vector representation. We have used glove.6B.300d as a word embedding which consists of words in vector representation that captures meaning in vector space. These beforehand converted vectors save time and are more accurate than the normal vectorization process and can have greater impact providing more meaningful vectors fed to the model.

c. *Models*

The model used in our project is Bi-directional

LSTM.Bi-directional LSTM

Bidirectional Long Short-Term Memory (BiLSTM) is a subset of the Long Short-Term Memory (LSTM) network architecture that is capable of processing sequences of data in forward as well as backward direction. Natural Language Processing (NLP) activities like speech recognition, machine translation, sentiment analysis, and text classification frequently make use of BiLSTMs.

A second LSTM layer that processes the input sequence in the reverse direction makes up the BiLSTM's design, which is identical to that of a conventional LSTM. The output of the forward LSTM layer is concatenated with the output of the backward LSTM layer, producing a final output that contains information from both directions of the input sequence. The key benefit of utilizing a BiLSTM is that it enables the network to understand the context of both the past and future for a particular input sequence. This is especially helpful in NLP jobs where the context of a word or phrase greatly affects its meaning.

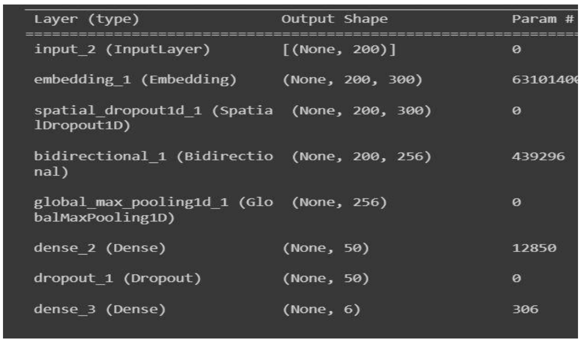
During training, the weights of the forward and backward LSTM layers are updated independently using backpropagation through time. The objective function of a The forward and backward LSTM outputs are often combined using a BiLSTM algorithm, such as their sum or their concatenation.

The model architecture consists of several layers for text classification. First, an input layer takes input sequences of maximum length (maxlen). Then, an embedding layer maps the words in the input sequences to dense vectors using pre- trained GloVe word embeddings, with the weights initialized from an embedding_matrix_glove. The embedding layer is set to be non-trainable to keep the pre-trained embeddings unchanged.

To prevent overfitting, a SpatialDropout1D layer applies a dropout rate of 0.2 to the embedded sequences. Next, a bidirectional LSTM layer with 128 hidden units is employed to capture contextual information from both past and future tokens in the input sequences. Dropout of 0.1 is applied during training, and recurrent_dropout of 0.1 is used during recurrent connections to regularize the LSTM layer.

A GlobalMaxPooling1D layer is then applied to the LSTM output, which extracts the most important features from the sequence representation. Subsequently, a dense layer with 50 units and ReLU activation function is used to learn higher-level features from the pooled representation. To further prevent overfitting, a dropout layer with a rate of 0.1 is added before the final output layer. A dense layer with a sigmoidal activation function is the final output layer that produces predicted probabilities for each class (6 classes in this case) in the input text. This model uses binary cross-entropy loss as the objective function and is compiled using the Adam optimizer. The performance of the model is evaluated using the area under the curve (AUC)metric.

Figure [2]: Bi-LSTM Architecture



```
Layer (type)                 Output Shape              Param #
================================================================
input_2 (InputLayer)         [(None, 200)]             0

embedding_1 (Embedding)      (None, 200, 300)          6310140

spatial_dropout1d_1 (Spatia  (None, 200, 300)          0
lDropout1D)

bidirectional_1 (Bidirectio  (None, 200, 256)          439296
nal)

global_max_pooling1d_1 (Glo  (None, 256)               0
balMaxPooling1D)

dense_2 (Dense)              (None, 50)                12850

dropout_1 (Dropout)          (None, 50)                0

dense_3 (Dense)              (None, 6)                 306
```

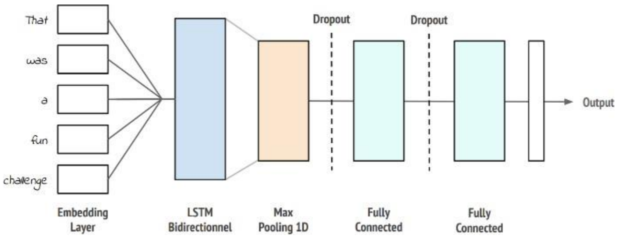


Figure [3]: Bi-LSTM Model

d. Metrics

To evaluate the performance of our algorithm, we go with AUC (area under the curve) and ROC (receiver operating characteristic), which are briefly discussed in this section.

AUC ROC

AUC ROC (Area Under the Receiver Operating Characteristic curve) is a widely used evaluation metric for classification tasks, including multiclass classification. It measures the ability of a model to distinguish betweenpositive and negative classes across all possible thresholds. For multiclass classification, the AUC ROC can be calculated by using a "one-vs-all" approach. This involves computingthe AUC ROC for each class against the remaining classes,resulting in a set of AUC ROC values. The final AUC ROCscore is then calculated as the average of these values.

The AUC ROC is preferred over other evaluation metrics, such as accuracy, when dealing with imbalanced datasets or when the cost of false positives and false negatives is not equal. It provides a more robust measure of a model's performance, taking into account both the true positive and false positive rate across all possible thresholds.

4. CONCLUSION AND FUTURE SCOPE

Toxic and harmful comments in social media have a number of detrimental effects on society. Therefore to accurately identify comments as toxic could provide many benefits and make social media a safe and toxicity freeplatform.

In this study, we developed a bi-directional LSTM model for the task of toxic comment classification. Our model achieved a remarkable accuracy of 98% using AUC ROC as the evaluation metric. Our results demonstrate the potential of deep learning models for accurately identifying toxic comments in online communities.

To achieve this high level of performance, we leveraged the power of the bi-directional LSTM architecture to captureboth forward and backward context information, allowing our model to effectively learn the complex relationshipsbetween words and sentences in toxic comments. Additionally, we utilized AUC ROC as the evaluation metric, which is a reliable measure for imbalanced datasets and betterrepresents the model's performance in real-world scenarios. Our study has several implications for the field of naturallanguage processing and online community management. Firstly, our model can be used as a valuable tool forautomating the identification of toxic comments and moderating online communities. This can improve the overallsafety and inclusivity of online spaces. Secondly, our study highlights the importance of leveraging deep learning techniques for achieving high accuracy in text classificationtasks.

In conclusion, our research provides a significant contribution to the field of natural language processing and online community management. The high accuracy of our bi-directional LSTM model, coupled with the use of AUC ROC evaluation metric, demonstrate the potential for deep learningmodels to effectively tackle the challenge of identifying toxic comments in online communities.

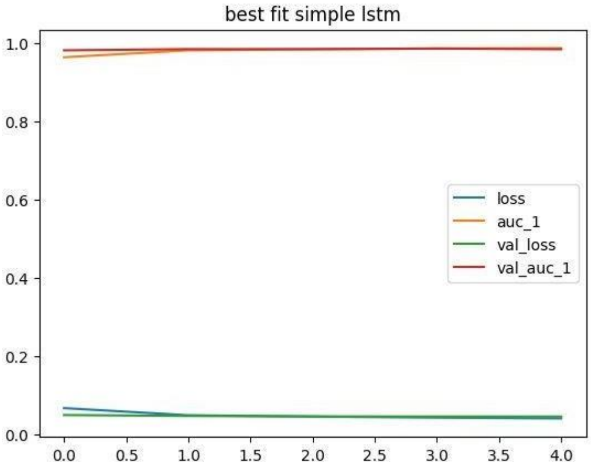


Figure [4]: Accuracy Report

ACKNOWLEDGMENT

We would like to express our gratitude to our research advisor Prof. Dr. P. Manikandan and our project coordinatorDr. M. John Basha for their constant guidance and support throughout the process.

References

[1] S. L. Blodgett and B. O’Connor, “Racial disparity in natural language processing: a case study of social media african-american English,” 2017, https://arxiv.org/abs/1707.00061.

[2] Onan, “Ensemble of classifiers and term weighting schemes for sentiment analysis in Turkish,” Scientific Research Communications, vol. 1, no. 1, pp. 1–12, 2021.

[3] Beutel, Z. Z. Chen, and E. H. Chi, “Data decisions and theoretical implications when adversarially learning fair representations,” 2017, https://arxiv.org/abs/1707.00075

[4] Friedler, J. Scheidegger, and V. S. Cii, “On the (im) possibility of fairness,”

[5] S. Korukoğlu and A. Onan, “Exploring the performance of instance selection methods in text sentiment classification,” in Artificial Intelligence Perspectives in Intelligent Systems, pp. 167–179, Springer, Cham, New York, NY, USA, 2016.

[6] Ms .Varsha Bairagi, Dr.Namrata Tapaswi 2016 Symposium on Colossal Data Analysis and Networking (CDAN)

[7] Pallam Ravi, Hari Narayana Batta, Greeshma S, Shaik Yaseen International Journal of Trend in Scientific Research and Development (IJTSRD) Volume: 3 | Issue: 4 | May-Jun 2019 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

[8] Neha Reddy1 , Neha Ram2 , Pallavi3 , Dr. K. Kranthi Kumar4 , Dr. K Sree Rama Murthy5 International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor.

[9] Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated Hate Speech Detection and the Problem of OffensiveLanguage. ICWSM.

[10] Fortuna, P., Nunes, C., & Sarmento, L. (2018). A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 51(4), 1-30.

[11] Park, S., & Fung, P. (2017). Learning to Classify Toxic Comments: Dataset Curation and Baseline Algorithm. arXiv preprint arXiv:1704.04579.

[12] Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019),75-86.

[13] Zhang, Y., & Wallace, B. (2018). Extracting Black-box Features via Minimal Causal Intervention. arXiv preprint arXiv:1805.1040.