Unlocking the Potential of NLP: CNN based Approaches for Contextual Hate Speech Detection

Nabila Islam Borno
dept. CSE
Brac University
Dhaka, Bangladesh
nabila.islam.borno@g.bracu.ac.bd

Ashfiqun Ahmed Miftah
dept. CSE
Brac University
Dhaka, Bangladesh
ashfiqun.ahmed.miftah@g.bracu.ac.bd

Atikur Rahman
dept. CSE
Brac University
Dhaka, Bangladesh
atikur.rahman1@g.bracu.ac.bd

Gazi Asif Imtiaz
dept. CSE
Brac University
Dhaka, Bangladesh
gazi.asif.imtiaz@g.bracu.ac.bd

Md Sabbir Hossain
dept. CSE
Brac University
Dhaka, Bangladesh
md.sabbir.hossain1@g.bracu.ac.bd

Mehnaz Ara Fazal
dept. CSE
Brac University
Dhaka, Bangladesh
mehnaz.ara.fazal@g.bracu.ac.bd

Annajiat Alim Rasel
dept. CSE
Brac University
Dhaka, Bangladesh
annajiat@bracu.ac.bd

***Abstract*—The research investigates the application of deep learning techniques, especially Convolution Neural Networks (CNNs), for detecting hate speech on social media platforms. Hate speech has increased significantly as a result of online communication's growing popularity, making it more challenging to maintain positive discourse. This study focuses on employing Natural Language Processing (NLP) methods to classify tweets into categories of hate speech, offensive language or neither. An analysis of a dataset consisting of English tweets is conducted to determine how well CNNs perform in accurately classifying content. The purpose of this project is to improve our knowledge of the processes that identify hate speech and to contribute to the creation of digital moderation systems that are more effective.**

***Index Terms*—Hate Speech Detection, Natural language processing, Hate Comments**

I. Introduction

The rapid expansion of social media platforms has given rise to an urgent challenge: The proliferation of hate speech. The identification and control of this type of harmful communication is of utmost importance due to its potential to cause extensive harm to both people and groups. To manage the volume and complexity of online interactions, traditional methods frequently depend on manual monitoring are insufficient.Convolutional neural networks (CNNs), one of the deep learning algorithms examined in this paper, have a lot of potential.in addressing this challenge.By leveraging the capabilities of Natural Language Processing (NLP), the research aims to develop a robust system capable of effectively categorizing tweets into hate speech, offensive language, or neither, thus contributing to safer online environments.

II. Literature Review

The success of NLP approaches is largely dependent on the quality of the dataset and the preprocessing techniques used. Tokenization and normalization were two essential preprocessing stages in the study that helped get the dataset ready for the CNN model. This is consistent with Waseem and Hovy's (2016) findings [1], which highlighted the significance of preprocessing and data quality in hate speech identification tasks.

The convolutional neural network (CNN) model was evaluated using metrics such as F1-score, recall, accuracy, and precision. The results of the investigation showed that although the model had a good accuracy rate overall, it had trouble accurately categorizing tweets as hate speech. This shows that more model parameter and training process optimization is required. The study of Schmidt and Wiegand (2017) [2] highlights the difficulties in obtaining high recall and precision and offers more insight into the complications of model assessment in hate speech detection.

The acknowledgment of the ethical implications of automated hate speech identification marks the study's conclusion. In building and implementing NLP models for such delicate jobs, ethical considerations—such as the possibility of prejudice and the effect on freedom of expression—are essential, as Fortuna and Nunes (2018) have shown. [3] Finally, by highlighting the advantages and disadvantages of CNNs, makes a substantial contribution to the current discussion on the identification of hate speech through the use of NLP and ML. Although the model has potential, further developments in this area will still require larger datasets, sophisticated model tweaking, and ethical concerns.

III. Proposed Methodology

In the Methodology section, the following key aspects are covered:

A. Data Collection

Gathering a Twitter dataset to identify hate speech was the first stage in our investigation. We made use of the CrowdFlower-distributed Hate Speech Identification dataset,

which included 24,783 English tweets divided into three categories: Hate, Offensive, and Neither.

B. Preprocessing

We will use the Python Keras library to create a preprocessing pipeline that will get the data ready for analysis. The following actions were necessary for this:
1. Removal of unnecessary characters
2. Reducing word inflections by stemming and converting to lowercase
3. Text tokenization into single words

C. Model Selection and Architecture

Considering the characteristics of hate speech identification, we choose to employ a Convolutional Neural Network (CNN) framework. The choice to employ a CNN was made due to its superior performance in text classification tasks, particularly when dealing with high-dimensional, variable-sized data sets like text documents.We are amazed by its own performance with text based data and decided to select this for hate speech detection.
We are examining a CNN architecture that draws inspiration from Yoon Kim's sentence categorization research. Layers include the input layer, embedding layer, pooling layers, convolutional layers with different filter sizes, and fully-connected layers make up the model. Based on default settings and discoveries from the literature, factors such as filter sizes, stride sizes, and pooling techniques were selected.

D. Performance Matrix

To analyze the model's performance, we'll use a range of evaluation metrics, including F1-score, accuracy, precision, and recall. We'll also try to use a confusion matrix to evaluate how well the algorithm classified tweets.

IV. EXPERIMENT

A. Data

Experiments were carried out using the implementation of the Hate Speech Identification dataset that was made available via CrowdFlower in order to evaluate the effectiveness of the suggestion. A total of 24,783 English tweets have been grouped into three distinct groups: Hate, which includes 1,430 tweets that include hate speech; Offensive, which includes 19,190 tweets that contain offensive language but do not contain any hate speech; and Neither, which comprises 4,163 tweets free of hate speech or abusive language. The following table provides an overview of the allocation of classes:

TABLE I
SUMMARY CLASSES

Class	# of Tweets
Hate	1430
Offensive	19190
Neither	4163
Total	24783

The table that follows provides an illustration of the distribution of tweets among the three categories. Notably, around five percent of the tweets include hate speech, whereas seventy-seven percent of the tweets contain language that is objectionable. Despite the fact that the dataset is rather modest in size, it is nonetheless used in this study. This is done in recognition of the imbalance in the number of tweets that belong to each class.

B. Pre Processing

During the preprocessing step, the content of each tweet was made normalized by using the Keras library in Python and making use of the Tokenizer class that it provides. This is the process that has been adopted:
1. Eliminating text from the tweet.
2. Word inflections may be reduced by using lowercase letters and stemming.
3. Creating tokens from the text.
Despite the fact that there are a variety of tools available for cleaning Twitter datasets, it is essential to point out that these tools were not taken into consideration in this work since the execution of these tools would have exceeded the constraints of the study.

V. EXPERIMENTAL SETUP

A. Data Splitting:

Because of the limited amount of time that we had available for the inquiry, we made the decision to divide the data into three distinct sets rather than selecting for cross-validation for the report that we generated for our study. Training a computationally intensive convolutional neural network (CNN) using cross-validation would need a significant amount of time. Consequently, the dataset was partitioned into training, validation, and test sets, with proportions of 50%, 30%, and 20%, respectively. The ratio was established by considering the overall size of the dataset, and it was concluded that a larger validation set was required for the purpose of evaluating the model and refining the parameters.

B. Architecture:

As previously stated, we are using CNN. Convolutional networks typically consist of three distinct types of layers: convolution, pooling, and fully connected. Training a whole convolutional neural network from the beginning is frequently impossible to operate and instead, pre-existing designs are typically adjusted. The study utilizes the CNN model, which adheres to the design suggested by Kim (2014). A non-linear convolution layer, a max-pooling layer, and a softmax layer are the components that make up this design. It is being predicted that Kim's design, which was first developed for the purpose of sentence categorization, would be adaptive to the problem of offensive or discriminatory language.

C. Input + Word Embedding:

The model accepts a tweet that has already been processed as its input, which is considered to be a string of words. Word embedding is also part of the model. Initialization of the embedding layer is accomplished by using the word2Vec word embedding, which is available to the public and has 300 dimensions.The embedding algorithm was trained on the news published by the Google dataset, consisting of three billion words, by employing a skip-gram model.

D. Hyperparameters:

The model parameters are mainly determined by default settings or previous empirical findings. The batch size and number of epochs are selected based on the training model, taking into consideration that the best results may vary based upon the data. Regardless of probable variances, the study showcases that utilizing these criteria produces commendable performances without the need for further modifications.

E. CNN:

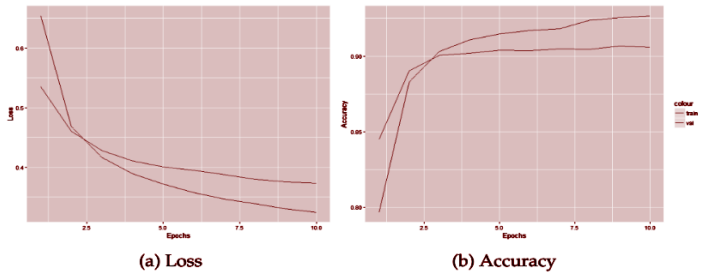
The word vector would have a dimension of d equal to 100. The result of this is an embedding layer that accepts an input feature space and a three-dimensional tensor of form (None, 100, 300). A two-dimensional convolutional layer is used to process the input. This layer is composed of filters of sizes 3, 4, and 5, and each filter has one hundred The embedding output is changed into a four-dimensional tensor shape so that the convolutional layers can be used. After that, the result is transformed into a four-dimensional tensor shape using max-pooling and embedding. Following that, the filters, each of which has its own layer, are merged together to form a single feature vector. Output that has been downsampled is subjected to max pooling, which results in the production of a feature vector. Then, this feature vector is sent into a softmax layer, which is used to make predictions about the probability distribution over all of the classes that are accessible.

F. Optimization:

The default parameters from Kim's (2014) study are used to train the model using the Adam algorithm, with the possibility of making adjustments to improve performance.The batch size is configured as 64, and the model is trained for 10 epochs. In Python,Tensorflow utilizes a 4-dimensional tensor for 2D convolution. The dimensions of this tensor correspond to batch, width, height, and channel.

VI. Results

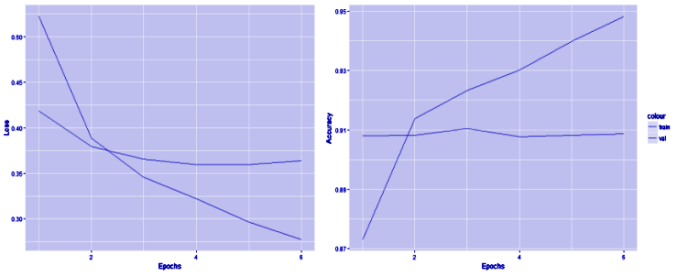
Once all the finishing touches have been added, the validation set must be used to evaluate the model's efficiency. After finishing this step, we will have a better grasp of whether or not making adjustments to the parameters is necessary in order to get the best possible performance. As was said before, Figure-1 provides a concise summary of the performance of the classifier when it was provided with the parameters that were described earlier.
The Adam technique is used to train the model, using the default values from Kim's (2014) research. However, there is the possibility that adjustments will be made in order to improve performance. The parameters for the batch size are set to 64, and the model is trained for a total of ten epochs. For two-dimensional convolution, Tensorflow in Python makes use of a tensor that is four-dimensional. With regard to batch, width, height, and channel, the dimensions of this tensor match to those dimensions.



(a) Loss (b) Accuracy

Fig. 1. The first model's loss and accuracy

The learning rate had to be slowed down by $\alpha = 10^{-4}$ so that the training model could be learned again with the same batch size and chance.
This was done on the basis of the observations that were made. The outcomes of this modification are shown in Figure Y, which demonstrates that the model profited from the slower learning rate.The training set's overall accuracy performance and that of the validation set are comparable, despite some overfitting on the training set. The accuracy performance remains consistent as a result.



(a) Loss (b) Accuracy

Fig. 2. Analysis of accuracy and loss after adjustments

The fact that this model was evaluated on a rather limited dataset is something that should be brought to your attention once again. Also, in order to prevent overfitting, it could be beneficial to gather additional data.

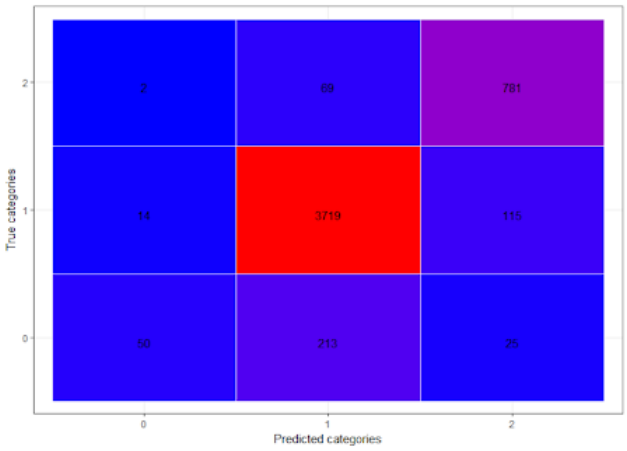


Fig. 3. Confusion Matrix

A. Final Results:

The model was able to correctly foresee each category with a loss of 36% and a 91% accuracy rate as a consequence of the parameters that were stated above. It was possible for the final model to achieve these outcomes, since it had an overall accuracy of 0.91, a recall of 0.90, and an F1-score of 0.90. Figure Z, on the other hand, reveals that the model did not properly designate a large number of tweets as hate speech tweets. This is shown by the way that the model is portrayed in the figure. In regard to fact, more than eighty percent of the individuals in the hate class were classified wrongly.
Considering the information presented in data , it is possible that this discrepancy might be assigned to not sufficient and imbalanced training data. It would indicate that the model has a predisposition for labeling tweets as offensive. Furthermore, due to the higher volume of tweets in that group, it did a better job of identifying the offending class, although it mistakenly classified some non-hate speech as hate speech.

VII. Discussion

A. Data:

Hate on Twitter may be expressed via three distinct methods: direct targeting towards an individual or a group, inside conversations among several users, or in a random manner without any specific target. In future research, it would be intriguing to examine the differentiation among the three manifestations of hatred shown by individuals on Twitter while analyzing hate speech. Further research might also prioritize the examination of the particular attributes and incentives of a user, for instance. Ideally, it would be preferable to gather more data in order to get a more precise differentiation between the performances of the models.

B. Ensemble:

There is a chapter dedicated to ensemble approaches. Prior research has used ensemble approaches and gained notable results in categorization challenges. Consequently, it is anticipated that these techniques may enhance the already achieved outcomes.

C. Limitations:

Convolutional neural networks are commonly used for image classification tasks, which include processing high-dimensional data, such as images. Although the architecture of a ConvNet is designed to reduce over-fitting, a substantial amount of data is typically required for a convolutional neural network to function optimally. Naturally, the quantity of data required is contingent upon the intricacy of the given work. In relation to the initial argument, convolutional neural networks may be excessive if the goal is really straightforward.Training a convolutional neural network can be a time-consuming process, particularly when dealing with extensive datasets. Typically, the training process can be accelerated by using specialized hardware, such as a GPU.Although convolutional neural networks (CNNs) exhibit translation invariance, they typically struggle to handle rotation and scale invariance unless explicit data augmentation techniques are employed.

D. Future Works:

A substantial number of tweets have been inaccurately categorized, particularly in the categorization of hate speech. Conducting an error analysis may provide valuable insights into the model's performance. When analyzing the inaccurate forecasts made by the hatred class, this may aid in understanding the challenges associated with predicting this class. It would be interesting to observe the use of certain phrases in differentiating between hate speech and offensive language.

VIII. Conclusion

In this particular piece of research, the objective was to identify instances of hate speech by using Natural Language Processing algorithms. It is essential to have a comprehensive understanding of the numerous meanings of hate speech since it is a phenomenon that is difficult to define. The evaluation of the appropriate research placed an emphasis on the use of deep learning models, in particular Convolutional Neural Networks (CNNs), for the determination of hate speech categorization tasks.
The application of the CNN model to a dataset of tweets that had been assigned with three classifications (hatred, foul language, and neither) produced encouraging results. According to Kim (2014), the CNN architecture that was used displayed a high level of efficiency with an accuracy rate of 91%. However, difficulties continued to exist, such as the incorrect classification of speech that did not demonstrate hatred as hate speech and the bias toward

offensive language that was caused by imbalances in the dataset.

CNNs have the potential to offer high results, even if the detection of hate speech continues to be a difficult problem. This potential becomes apparent with datasets that are richer and bigger. Due to the subjective nature of hate speech categorization, it is essential to have datasets that are both stable and varied in order to get correct findings.

REFERENCES

[1] Waseem, Z., Hovy, D. (2016). "Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter."

[2] Schmidt, A., Wiegand, M. (2017). "A Survey on Hate Speech Detection using Natural Language Processing."

[3] Fortuna, P., Nunes, S. (2018). "A Survey on Automatic Detection of Hate Speech in Text."

[4] Soujanya Poria Tom Youngy, Devamanyu Hazarikaz and Erik Cambria. Recent trends in deep learning-based natural language processing. 2018"

[5] iqi Zhang.(2018) "Hate speech detection: A solved problem? the challenging case of the long tail on Twitter. "

[6] Schmidt, A., Wiegand, M. (2017). "A Survey on Hate Speech Detection using Natural Language Processing."

[7] Packt Publishing. Sentence classification using cnns. https://www.datasciencecentral.com/profiles/blogs/sentence-classification-using-cnns, July 10, 2018."

[8] Daniel Jurafsky and James H. Martin. Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition. 2017.

[9] Sean McElwee. The case for censoring hate speech. https://www.alternet.org/civil-liberties/case-censoring-hate-speech,2013.

[10] Avid Robinson, Ziqi Zhang, and Jonathan Tepper. Hate speech detection on twitter: Feature engineering v.s. feature selection. 2018.