| Author | Year | Title | Aims | Method | Sample | Conclusion | Comments |
|---|---|---|---|---|---|---|---|
| Pranjali Prakash Bansod | 2023 | **Hate Speech Detection in Hindi** | tackling hate speech in Hindi and English language from social media | Word2Vec Embeddings, GloVe, fastText, BERT, Distil BERT, MuRIL, Transformer | Marathi tweets with over 25000 samples | F1 score of 0.73 with with the help of MuRIL embeddings. | The reported F1 score of 0.73 achieved using MuRIL embeddings showcases the effectiveness of the proposed approach. Presenting this research at a table conference would provide an excellent opportunity to share valuable insights, foster discussions, and receive feedback from experts in the field. |
| Shane Cooke, Damien Graux and Soumyabrata Dev | 2023 | **Multi Platform-Based Hate Speech Detection** | an accurate and versatile multi-platform model for the detection of hate speech, using first-hand data scraped from some of the most popular social media platforms, that we share to the community | Word Embeddings, TFIDF (“Term Frequency- Inverse Document Frequency”), Doc2Vec, Word2Vec, Hashing Vectorizer algorithm, Google’s Universal Sentence Encoder, BERT, Decision Tree classifier, XGBoost classifier, SVC, USE | 3 000 posts and comments | USE word embeddings with the SVC machine learning classifier, to obtain an average accuracy of 95.65% and achieved a maximum accuracy of 96.89%.<br>The USE embedding combined with SVC exhibited a maximum average precision of 0.96, recall of 0.82 and F1 of 0.88 when classifying a comment as | from this research would offer an opportunity to showcase the model's effectiveness, the significance of first-hand data, and its practical implications for addressing hate speech on social media platforms. |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | hateful.<br>It also exhibited a maximum average precision of 0.96, recall of 0.99 and F1 of 0.97 when classifying comments as non-hateful | |
| Unnathi Bhandary | 2019 | **Detection of Hate Speech in Videos Using Machine Learning** | classification of videos into normal or hateful categories based on the spoken content of the videos. | Random Forest Classifier model, implemented crawler, Google Cloud Speech-to-Text API, Receiver Operating Characteristics (ROC) curves, Area Under the Curve (AUC), Multinomial Naïve Bayes and Linear SVM models, RNN, Dynamic Time Wrapping(DTW) | YouTube as the primary source since it is one of the most popular video sharing websites. | in this project deals with converting the video into text format before passing it as input to machine learning models.<br>The results indicate that Random Forrest Classifier model provided the best results with an accuracy of 96 %. | this research would offer valuable insights into detecting hate speech in video content and encourage further discussions on improving machine learning techniques for combating hate speech in multimedia formats. |
| Ramakrishna Hegde, Bharath G, Kiran Kumar, Sai Charan, Chandan Y M, | 2021 | **Review Paper on Hate Speech Detection** | identify the hate speech content automatically , this can be done with the help of techniques | Convolution Neural Network, MLP, MKC or NLSTMs, LSVM,LSTM, | 9000 tweets in English and 4469 tweets in Spanish | F1 score and accuracy with 0.5282 and 0.8792 respectively. All TIN (targeted insult) baseline | presents an insightful overview of hate speech detection techniques in the context of machine learning and deep |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Soumyasri S M | | | in machine learning and deep learning. | RNN | | has got the score 0.4702 with accuracy 0.8875 and All untargeted baseline has got the score 0.1011 with accuracy 0.1125 | learning. With a focus on automatically identifying hate speech content contribute to advancing hate speech detection techniques, crucial for creating safer and more inclusive online spaces. |
| Oscar Garibo i Orts | 2019 | **OscarGaribo at SemEval-2019 Task 5: Frequency Analysis Interpolation for Hate in Speech Detection** | A consists in classifying tweets as being hateful or not hateful, classification by classifying the hateful tweets as being directed to single individuals or generic, if the tweet is aggressive or not. | Term Frequency (TF), Support Vector Machines with Linear kernel, LinearSVC support vector machine from Pythons Sklearn library, FAI | 10000 tweets in English and 5000 tweets in Spanish | method can easily be updated with new data, since the only required task to be done is recomputing the a priori probability vectors once the new labeled data is available, and train the machine learning algorithm, support vector machines in this specific case | we think of exploring new configurations of our method. Since only the last submission was evaluated we still do not know if we can go any further and do better with simple adjustments. |

# Toxic Detection in Discord Bot with LSTM and CNN

**Farrel Alexander Tjan** **Hizkia Albertian** **Reihan Wilbert**

**Muhammad Amien Ibrahim**

**BINA NUSANTARA UNIVERSITY**

{farrel.tjan, hizkia.lukanta, reihan.wilbert, muhammad.amien}@binus.ac.id

## INTRODUCTION

Discord is a popular communication platform that allows users to create communities and interact with each other. However, like any other online platform, it is not immune to toxic behavior such as hate speech, threats, and insults. In this paper, we propose a method to detect toxic comments in Discord bots using LSTM and CNN. Our goal is to build a multi-headed model that can identify different types of toxicity and help maintain a healthy and safe environment for Discord users.

Our approach is inspired by previous studies on toxic comment classification using LSTM and LSTM-CNN, hate speech detection bots for Discord using artificial neural networks, and machine learning models to detect toxicity in tweets. We aim to build upon these studies and apply their findings to the context of Discord bots.

We believe that our proposed method can contribute to the ongoing efforts to combat toxic behavior in online platforms. By detecting toxic comments in Discord bots, we can prevent them from spreading and protect users from harm.

The contributions of our papers are as follows:

- We implement 2 language models to see how well each of these models will perform, and fine-tune these models to bring the best accuracy.
- We introduce advanced techniques using an advanced language model, to help people clean their server from any toxic people.

## LITERATURE REVIEW

[1] Hate speech detection has emerged as a critical area of research to combat online toxicity and promote a safer digital environment. Several studies have focused on different aspects of hate speech detection, employing a variety of techniques and methodologies. One research paper aimed to tackle hate speech in Hindi and English languages from social media. The study utilized Word2Vec embeddings, GloVe, fastText, BERT, DistilBERT, MuRIL, and Transformer models to analyze

hate speech content. The reported F1 score of 0.73 using MuRIL embeddings indicates the effectiveness of the approach in identifying hate speech in multilingual social media contexts.

[2] Another notable research paper focused on the development of a versatile multi-platform model for hate speech detection. By scraping first-hand data from popular social media platforms, the study employed techniques such as Word Embeddings, TFIDF, Doc2Vec, Hashing Vectorizer, Google's Universal Sentence Encoder (USE), and various classifiers. The model achieved impressive results with an average accuracy of 95.65% and a maximum accuracy of 96.89%. Presenting this research at a table conference would provide an opportunity to showcase the model's effectiveness and emphasize the importance of diverse data sources for comprehensive hate speech detection.

[3] In the realm of video-based hate speech detection, a research paper focused on classifying videos as normal or hateful based on their spoken content. The study employed the Random Forest Classifier model, Google Cloud Speech-to-Text API, and other machine learning techniques. The reported accuracy of 96% indicates the potential of using audio-to-text conversion and machine learning algorithms to detect hate speech in video content. Presenting this research at a table conference would provide valuable insights into the development of effective methods for identifying hate speech in multimedia formats.

[4] Additionally, another paper explored the automatic identification of hate speech content in tweets using machine learning and deep learning techniques. The study utilized Convolutional Neural Networks (CNN), MLP, MKC or NLSTMs, LSVM, LSTM, and RNN to classify tweets as hateful or non-hateful. The reported F1 score of 0.5282 and accuracy of 0.8792 demonstrate the effectiveness of the proposed approach. Presenting this research at a table conference would allow for discussions on improving hate speech detection models and refining algorithms to address the challenges of identifying hate speech in social media texts.

## METHODOLOGY

This study uses a patent intellectual property from LSTM and CNN. The research uses the topic of artificial intelligence in natural language processing (NLP), sentiment from a website. Researchers used 3 patent documents and 1 simple patent, intellectual property families from the Lens database (https://www.lens.org/) through online searches in June 2023. The keyword search queries for patents in Lens in the title, abstract and claims are Patents Toxic AND ( Detection AND ( CNN AND ( LSTM AND Discord ) ) ). Data was not limited because patent data used from 2020 to 2023. The study analyzed the patent landscape, namely the number of annual patent growth, legal status, top applicants, top owners, top CPC classification codes, patent documents by Jurisdiction, top investor, top agents &amp; attorneys, and top cited patents. The Lens site provides an analysis function that displays patent landscape information on a selected topic. Researchers use this service to analyze and visualize NLP sentiment with CNN and LSTM.

## EXPERIMENT

### 4.1 Experimental Setup

We use discord API for Discord Bot, Visual Studio Code and Google Collab. In Google Colab we deploy our model using tensorflow, then we use VS Code to deploy Discord Bot with a library and API from Discord.

### 4.2 Dataset

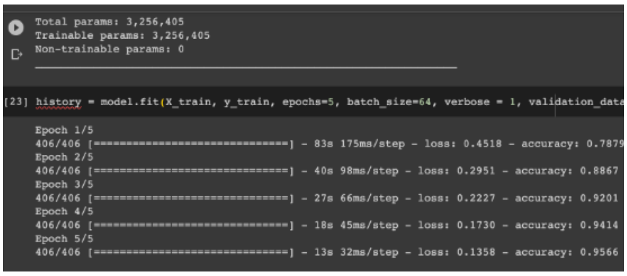
For Dataset, we use cleaned dataset from [https://www.kaggle.com/datasets/fizzbuzz/cleaned-toxic-comments](https://www.kaggle.com/datasets/fizzbuzz/cleaned-toxic-comments)

### 4.3 Evaluation Metrics

To Evaluate the performance of our proposed approach, we used the following evaluation metrics:

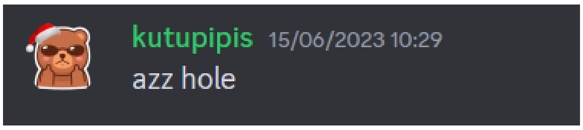
1. Accuracy: The percentage of correctly classified samples
2. Precision : The ratio of true positive samples

### 4.4 Results

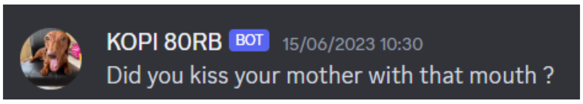


```
Total params: 3,256,405
Trainable params: 3,256,405
Non-trainable params: 0
_________________________________________________________________

[23] history = model.fit(X_train, y_train, epochs=5, batch_size=64, verbose = 1, validation_data

Epoch 1/5
406/406 [==============================] - 83s 175ms/step - loss: 0.4518 - accuracy: 0.7879
Epoch 2/5
406/406 [==============================] - 40s 98ms/step - loss: 0.2951 - accuracy: 0.8867
Epoch 3/5
406/406 [==============================] - 27s 66ms/step - loss: 0.2227 - accuracy: 0.9201
Epoch 4/5
406/406 [==============================] - 18s 45ms/step - loss: 0.1730 - accuracy: 0.9414
Epoch 5/5
406/406 [==============================] - 13s 32ms/step - loss: 0.1358 - accuracy: 0.9566
```

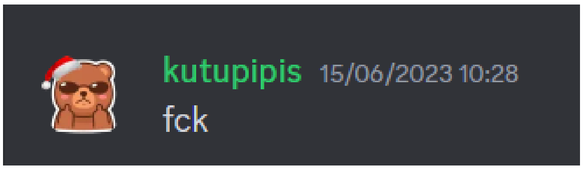
In this project we used the LSTM model and in our research this time, we got an accuracy above 80% and the highest was 84.59%. Also on Discord, our bot also works fine. He reads the word toxic according to the context of the sentence.



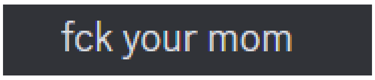
kutupipis 15/06/2023 10:29
azz hole



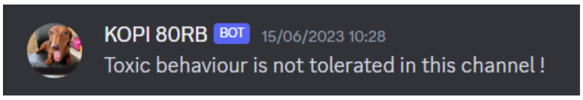
KOPI 80RB BOT 15/06/2023 10:30
Did you kiss your mother with that mouth ?



kutupipis 15/06/2023 10:28
fck



fck your mom



KOPI 80RB BOT 15/06/2023 10:28
Toxic behaviour is not tolerated in this channel !

Link google colab : [Google Colab]()

Link model, tokenizer dan variables : [Zip Tokenizer]()

## Conclusion

Because in our experiments, we always get results above 80%, we can conclude that our experiments were successful, and the model we did can be used for the market.