IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

COGNITIVE METHOD TO DETECT TOXIC COMMENTS IN SOCIAL MEDIA

[1]Jomy Joseph, [2]Sneha Rose, [3]M.Jayakumar
[1]Student, [2]Assistant Professor, [3]Assistant Professor
[1]M.Sc Data Science and Business Analysis,
[1]Rathinam College of Arts and Science, Coimbatore, India

Abstract : Social media is considered the most important activity that helps to consume more experience and knowledge. This is due to the massive increase in internet users worldwide, which continues to increase from millions to billions using the Internet in the operation of social media. Due to these, it has become the first information that the user consumes after waking up and before sleeping. It helps in the fulfilment of small moments of our day-today lives. Toxicity in online comments has become a major concern in today's digital world, and detecting such comments automatically can greatly improve the safety and quality of online discussions. The project is developed in order to produce a model which can detect toxic comments with the use of Hybrid Neural Network (HNN), which is a combination of Convolutional Neural Network (CNN) and Recurrent Neutral Network (RNN). The results indicates that the use of Hybrid Neural Network has resulted in an increased efficiency in detecting toxic comments. This project mainly focuses on creating a model that can detect and control various toxics such as threats, obscenities, insults and identity-based hatred in an automated manner. This model can be further incorporated in web applications, social media, blogs, etc.

IndexTerms **- Toxic Comment, Natural Language Processing**

I.INTRODUCTION

Toxic comment detection using deep learning has emerged as a crucial application in the realm of natural language processing (NLP), aimed at identifying and filtering out harmful or offensive content in online discussions. Deep learning models, particularly neural networks, have shown remarkable effectiveness in automatically detecting toxic comments across various digital platforms. These models typically leverage architectures such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), including variants like long short-term memory (LSTM) or gated recurrent units (GRUs), to process and analyze text data. Through the utilization of large-scale labeled datasets, deep learning algorithms can learn intricate patterns and semantic nuances associated with toxic language, enabling them to distinguish between benign and harmful comments with high accuracy.

The process involves preprocessing textual data, tokenizing words, and representing them as numerical vectors, which are then fed into the deep learning model for training. During training, the model adjusts its parameters iteratively to minimize the prediction error, effectively learning to recognize toxic patterns in the data. Techniques such as transfer learning, where pre-trained models are fine-tuned on specific toxic comment detection tasks, have also proven effective in improving model performance, especially when labeled data is limited. The deployment of deep learning-based toxic comment detection systems has significant implications for online community moderation, content filtering, and maintaining a safer digital environment by promptly identifying and mitigating instances of toxicity.

On social media, any user can post comments since social media masks the identity of the user. So, the rate of toxic comments is increasing day by day, and it leads to disturbance in the victim's mental health. Currently social media does not provide any features to detect toxic comments and report the user. A research study shows that most of the internet users who have been victims of online harassment and experienced some kind of harassment. So, its high time to develop a model to automatically detect and remove toxic comments from social media discussion forums to ensure the healthy use of these platforms[1].

Automatic detection and removal of toxic comments helps the social media provider and users to detect and delete toxic comments. This helps the social media providers to analyses the users who are posting toxic comments and also helps in reporting those users. Much faster and less energy-consuming. Manual detection of toxic comments takes time and also its a tedious task. This model helps to detect the comments faster and in an automated manner. More efficient than the exciting model. Presently social media does not provide any feature to detect and delete these comments automatically. The present method of removing these comments manually is a tedious task. The new method helps in automatic removal of toxic comments and more efficient than existing model. Efficient method for multilingual comments. This model works better with regional languages too. This helps in converting comments from regional language to English and then detects the toxicity of comments.

II.RESEARCH METHODOLOGY

2.1 Existing System

In the 21st century, social media is considered the most important activity that helps to consume more experience and knowledge. This is due to the massive increase in internet users worldwide, which continues to increase from millions to billions using the Internet in the operation of social media. Due to these, it has become the first information that the user consumes after waking up and before sleeping. It helps in the fulfillment of small moments of our day-to-day lives. This platform has already faced a great deal of criticism due to the proliferation of hate speech, threats, insults, and libelous activities, which makes it difficult for many internet service providers to control. Any user can post comments since social media masks the identity of the user. Therefore, the rate of toxic comments is increasing day by day and leads to disturbance in the victim's mental health. The existing method to remove such comments is by removing them manually by the host/content-creator [5]. The primary goal of the project is to build a model under the domain of social media to detect and filter out toxic comments from the social media platforms. This model helps in detecting comments automatically, this reduces the time of detecting such comments and also reduce the effort of users/provider.

Constructing a model within the realm of social media with the ability to identify various forms of toxicity such as threats, obscenity, insults, and hate based on identity, as found in social discussion forums.The primary goal of the proposed model is to develop an effective method to automatically detect and remove toxic comments from social discussion forums to ensure the healthy use of these platforms[3].

The number of malicious people has increased as well, owing to the unexpected advent of the web-using public. These days, keeping the discussions positive and inclusive is the main responsibility of all online platform providers. The primary duty of the content creator (the host) is to filter out these comments in order to prevent the spread of negativity or hatred among people, since these are publicly accessible to the public, who are viewed from a variety of social groups, age groups, communities, and socioeconomic backgrounds[2]. This initiative assists in removing offensive content and maintaining a toxic-free environment on social media platforms[4]. The toxic comment detection model can be applied in social media platforms and online discussion forums. The model can be used by the users and social media providers to keep their website or discussion forums toxic free[4]. This can be applied in many applications and websites to make their platform toxic free and also helps the users to use those platforms more conveniently and efficiently.

III.PROPOSED WORK

3.1 System Design

A system design serves as the conceptual framework outlining its structure, functionality, and various perspectives. The Object Detection system employs a three-tier architectural design. The user will use a web application on a browser to send http requests to a web server hosted using Flask framework. The API endpoints created using flask will interact with the database server such as MySQL. It gives a response back to the web server which in turn respond back to the client application[10].

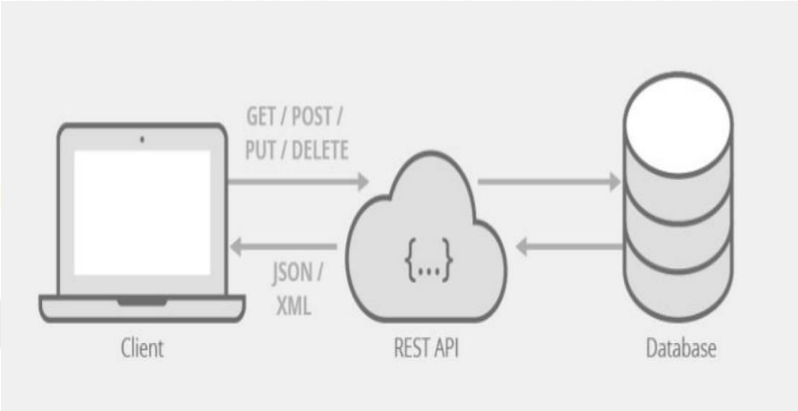

Fig.1 System Architecture

The web application contains 5 modules. The modules are Sign Up/Register, Input comment, Toxic comment, Identification of LOT and Displaying Result. The following section gives detailed information about the functionalities of the different modules in the application.

- **Sign Up/Register:** It contains name, email address, and an OTP-based authenticator. The main aim is to authenticate the user who is posting their comment.

- **Input Comment:** Here we accept the comments given by the users which will be later used for processing.

- **Toxic Comment:** This algorithm classifies the comments into toxic and nontoxic comments and detects the level of toxicity(LOT).

- **Identification of LOT:** This step decides whether a comment should be posted or removed/reported. If the level of toxicity is more than 5comment will be removed as well as reported. Otherwise, it will be posted in the comment section.

- **Displaying Result:** If the LOT is less than 5comment. Else a report message will pop up.

3.2 Implementation

- **Data collection and pre-processing:** The first step is to gather a dataset of comments that are labelled as toxic or non-toxic. The dataset may be obtained from various sources such as social media platforms, discussion forums, or online communities. The data is then preprocessed to clean and normalize the text, which may involve tasks such as removing punctuation, stop words and converting text to lowercase.

- **Feature extraction:** Next, the preprocessed text is transformed into numerical representations or features that can be used as inputs to the deep learning model. Common feature extraction techniques for text include bag of words, TF-IDF, and word embeddings.

- **Deep learning model:** The heart of the system is a deep learning model that learns to classify comments as toxic or non-toxic based on the extracted features. Common architectures for this task include convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers. Model was build using deep learning techniques like CNN and RNN. Rather than commencing with thoroughly established requirements, the approach involves implementing a set of software requirements followed by testing, evaluation, and the identification of additional requirements [1]. Furthermore, a hybrid neural network (HNN) model, which integrates both approaches, has been developed.

- **Training:** The model is trained using the labeled dataset, where the input is the preprocessed text and the output is the corresponding toxicity label. During the training process, the model's parameters are optimized to reduce a loss function, which quantifies the disparity between predicted and actual labels.

- **Evaluation:** Following the training phase, the model undergoes evaluation using an independent test dataset to gauge its performance based on metrics such as accuracy, precision, recall, and F1-score.

- **Deployment:** Upon completion of training and evaluation, the model becomes ready for deployment within a production environment, enabling it to autonomously classify incoming comments as either toxic or nontoxic in real-time. Deployment may entail integrating the model into an existing web application, API, or chatbot.

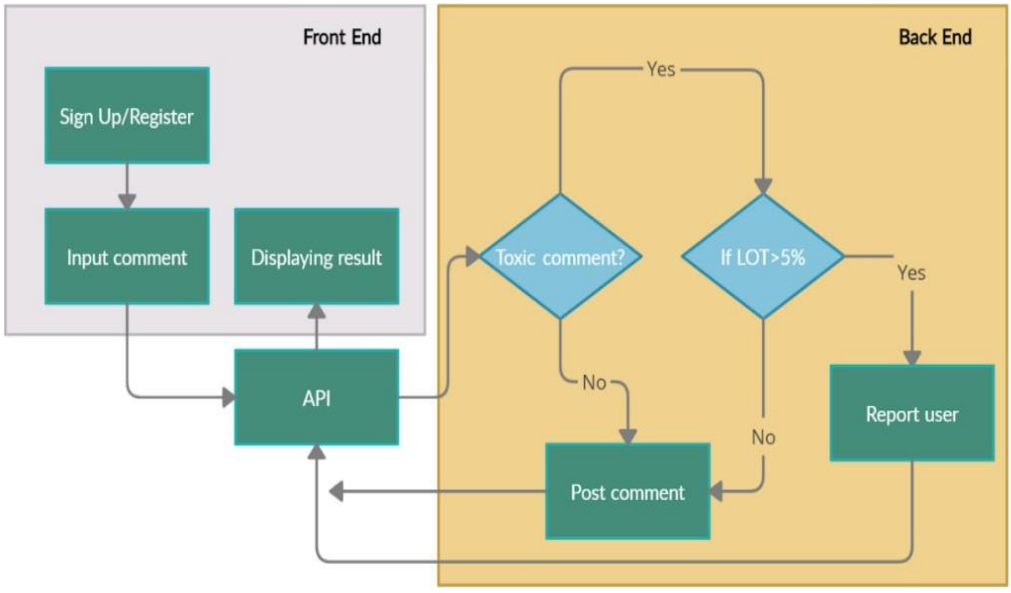

Fig.2 Flow Diagram of the Deep Learning Model

3.3 Algorithms

3.3.1 Recurrent Neural Networks(RNN)

Recurrent Neural Networks (RNNs) represent a pivotal advancement in the realm of artificial intelligence and machine learning, particularly in the domain of sequential data processing. Unlike traditional feedforward neural networks, RNNs are designed to capture temporal dependencies and maintain a form of memory across input sequences. This unique architecture allows RNNs to excel in tasks involving sequential data, such as time series analysis, natural language processing, and speech recognition[9].

The hallmark feature of RNNs is their ability to exhibit dynamic temporal behavior, making them well-suited for applications where context and order are crucial. Each node in an RNN is equipped with a hidden state that serves as a form of memory, enabling the network to retain information about previous inputs. However, standard RNNs face challenges in capturing longterm dependencies due to the vanishing and exploding gradient problems.

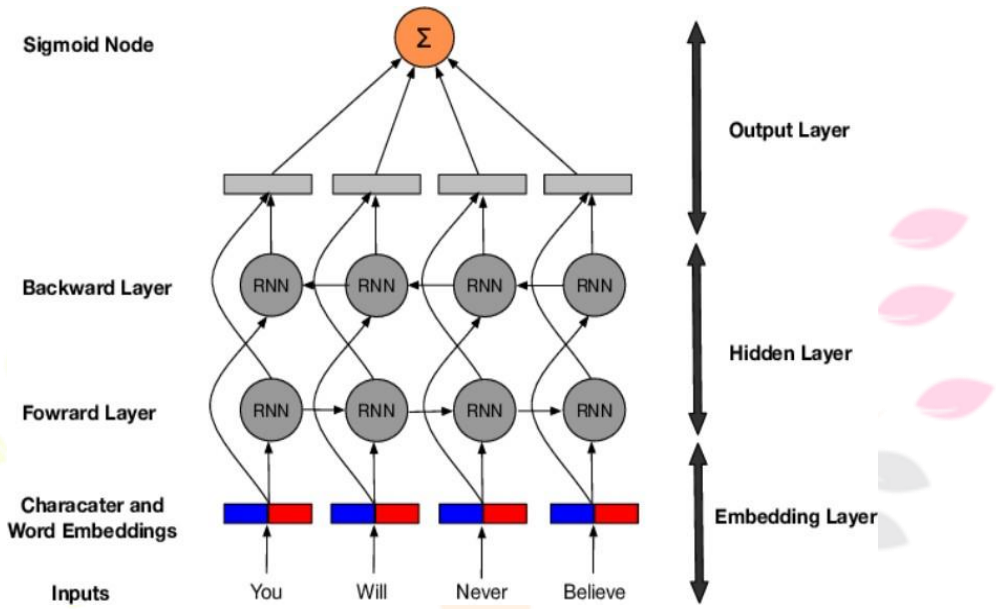

Fig.3 RNN Structure

3.3.2 Convolutional Neural Network(CNN)

Convolutional Neural Networks (CNNs) have become fundamental in computer vision, transforming tasks related to image processing and pattern recognition. CNNs are specifically designed to excel in tasks where the spatial arrangement of features is crucial, making them highly effective for image classification, object detection, and segmentation[11]. The fundamental building blocks of CNNs are convolutional layers, which systematically scan input data with learnable filters, capturing hierarchical representations of features. This characteristic enables CNNs to automatically learn and identify complex patterns, making them adept at recognizing objects invariant to translation, rotation, and scaling.

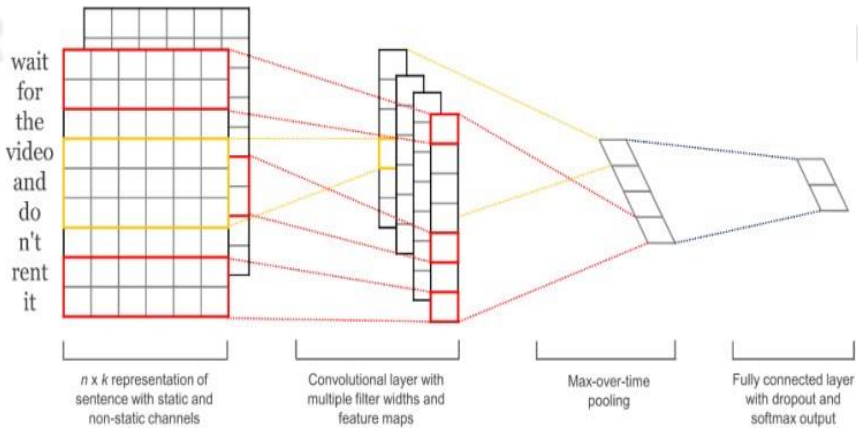

Fig.4 CNN Structure

One key advantage of CNNs is their ability to reduce the number of parameters by sharing weights across the convolutional filters, making them computationally efficient compared to fully connected networks. Max-pooling layers further contribute to this efficiency by downsampling feature maps, retaining essential information while reducing computational complexity. Transfer learning, facilitated by pre-trained CNN models on large datasets like ImageNet, has further accelerated the adoption of CNNs in various applications, allowing developers to leverage knowledge gained from one task for improved performance in another[12]. As a result, CNNs have become instrumental in numerous domains, from medical image analysis to autonomous vehicles, solidifying their status as a cornerstone technology in the modern era of artificial intelligence.

3.3.3 Long Short-Term Memory (LSTM)

The Long Short-Term Memory (LSTM) architecture belongs to the category of recurrent neural networks (RNNs) and is specifically engineered to address the challenge of the vanishing gradient problem commonly observed in conventional RNNs.Developed by Sepp Hochreiter and Jürgen Schmidhuber in 1997, LSTM networks have gained significant popularity due to their ability to effectively capture long-term dependencies in sequential data. In contrast to standard RNNs that face difficulties in preserving information across lengthy sequences, LSTM networks employ distinctive memory cells equipped with input, output, and forget gates to manage the information flow within the network. These gates, composed of sigmoid and element-wise multiplication functions, control the extent to which information is stored, retrieved, or discarded at each time step, enabling LSTMs to maintain information over extended periods. This capability makes LSTMs particularly well-suited for a wide range of tasks involving sequential data, including natural language processing, speech recognition, time series prediction, and more. . By addressing the challenges associated with modeling long-term dependencies, LSTM networks have become a cornerstone in the field of deep learning, offering robust solutions for tasks requiring sequential data processing.

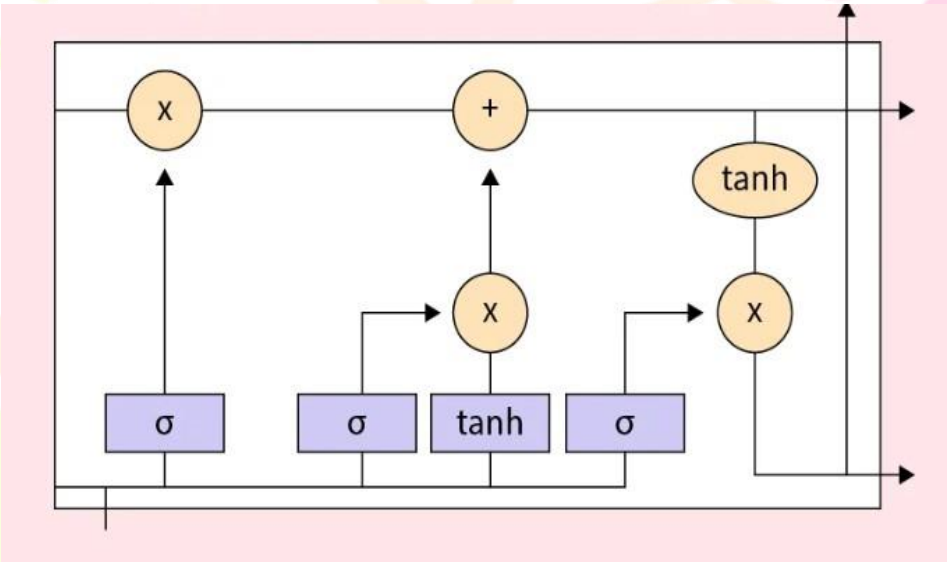

Fig.5 LSTM Structure

IV. RESULTS AND DISCUSSION

The fundamental challenge of text classification in natural language processing plays a pivotal role in various applications, such as sentiment analysis, topic categorization, online searching, and information filtering. Artificial Intelligence (AI) and Deep Learning techniques can be employed alongside NLP for constructing and training the model for text classification [8].

Epochs	RNN	CNN	LSTM
3	0.9944	0.9904	0.9840
4	0.9944	0.9904	0.9870
5	0.9944	0.9939	0.9877

Table 1: Accuracy of the system architecture in %

In this system, the user inputs a comment. Then the system verifies whether the given common is toxic or not. If the comment is non-toxic then, the comment will be displayed on the website. In case the comment is identified as toxic then, the comment will be sent for identification of toxicity level. If LOT is less than 58 else the system will remove the given comment and report the user for the same The accuracy of the model in RNN is 0.9944,in CNN 0.9904,in LSTM 0.9877.The RNN model gives more accuracy than CNN and LSTM.

V.CONCLUSION

Toxic comment identification using deep learning algorithms is a major advancement in the fight against the ubiquitous problem of online toxicity. By using this creative strategy, we have seen impressive advancements in the capacity to recognize and filter dangerous information on a variety of digital platforms. Recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer-based designs such as BERT are examples of deep learning models that have demonstrated effectiveness in accurately and reliably recognizing damaging statements.

The continuous evolution and refinement of these deep learning models, coupled with the availability of vast amounts of labeled data, have contributed to enhancing their robustness and adaptability to diverse linguistic nuances and contextual variations. However, challenges persist, including the need for large-scale annotated datasets encompassing diverse languages and cultural contexts, as well as the ongoing cat-and-mouse game with malicious actors who strive to evade detection mechanisms.

REFERENCES

[1] Revati Sharma , Meetkumar Patel, "Toxic Comment Classification Using Neural Networks and Machine Learning", Vol. 5, Issue 9, September 2018, DOI 10.17148/IARJSET.2018.597,pg no:47- 52.

[2] NavoneelChakrabarty, "A Machine Learning Approach to Comment Toxicity Classification", 2016.

[3] Pallam Ravi, Hari Narayana Batta, Greeshma S, Shaik Yaseen, "Toxic Comment Classification", Volume: 3, Issue: 4, 2019.

[4] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation.

[5] Georgakopoulos, V., Tasoulis, S., Vrahatis,A., Plagianakos, P., (2018). Convolutional Neural Networks for Toxic Comment Classification, ACM Proceedings of the 10th Hellenic Conference on Artificial Intelligence, pp. 35.

[6] Salvatore Carta, Andrea Corriga, Riccardo Mulas, Diego Reforgiato Recupero and Roberto Saia, "A Supervised Multi-class Multi-label Word Embeddings Approach for Toxic Comment Classification",2019.

[7] Dos Santos, Cicero & Gatti de Bayser, Maira. (2014). Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts.

[8] Pooja Parekh, Hetal Patel, "Toxic Comment Tools: A Case Study", Volume 8, No. 5, May-June 2017, pg no: 964 – 967.

[9] Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving Google's Perspective API Built for Detecting Toxic Comments. arXiv preprint arXiv:1702.08138 (2017).

[10] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep Convolutional Networks on Graph-Structured Data. CoRR abs/1506.05163 (2015). arXiv:1506.05163 http://arxiv.org/abs/ 1506.05163.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In Advances in Neural Information Processing Systems 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3111–3119.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient based learning applied to document recognition. Proc. IEEE 86, 11 (Nov 1998), 2278– 2324.

[13] Ge Song, Yunming Ye, Xiaolin Du, Xiaohui Huang, and Shifu Bie. 2014. Short Text Classification: A Survey. Journal of Multimedia 9, 5 (2014).

[14] B. Herger, "Detecting toxic comments with multi-task deep learning," 2018. https://www.hergertarian.com/ detecting-toxic comments.

[15] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. J. Mach. Learn. Res. 12 (Nov. 2011), 2493–2537. http://dl.acm.org/citation.cfm?id=1953048. 2078186.