“HATE SPEECH DETECTION USING MULTI-CHANNEL CONVOLUTIONAL NEURAL NETWORK”

A DISSERTATION

Submitted in partial fulfilment of the requirements for the award of the degree
of
Master of Technology (MTech)
in
Computer Science and Engineering (CSE)

Submitted by
T Akhilesh Naidu (Roll No: **2K19/CSE/22**)

under the guidance of
Dr. Shailender Kumar
Associate Professor
Dept. of Computer Science and Engineering

DEPT. OF COMPUTER SCIENCE AND ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY, DELHI
DECEMBER, 2020

DECLARATION

"I, T Akhilesh Naidu, understudy of M. Tech. (Software engineering) announce that the undertaking disquisition named **"Hate Speech Detection utilizing Multi-Channel Convolutional Neural Network"** which is put together by us to the Department of Computer Engineering, Delhi Technological University, in fractional satisfaction of the prerequisite for the honour of the level of Master of Technology is a unique work and has not been replicated from any source without legitimate reference. This work has not recently framed the reason for the honour of any Fellowship, Degree, Diploma Associateship, or other comparable title or acknowledgment".

New Delhi
29th-06-2021

T Akhilesh Naidu

CERTIFICATE

This is to confirm that the Project Dissertation named "**Hate Speech Detection using multi-channel Convolutional Network"** which is presented by T Akhilesh Naidu of Computer Science and Engineering, Delhi Technological University Delhi in fractional satisfaction of the prerequisite for the honor of the level of Master of Technology is a record of the undertaking work did by the understudies under my watch. Supposedly, this work has not been submitted partially or full for any Degree or Diploma to this University or somewhere else.

Dr. SHAILENDER KUMAR
SUPERVISOR
Associate Professor
Department of Computer Engineering
Delhi Technological University

ACKNOWLEDGEMENT

"I might want to show my appreciation and thankfulness to my tutor and task direct,
Dr. Shailender Kumar, Associate Professor, Delhi Technological University for giving us the chance and the necessary rules to take a shot at this task alongside various counsels. I value my seniors for their thoughtful collaboration and significant consolation that assisted us with finishing this mission. I additionally offer our thanks to all other employees of our specialty for their steady consolation, and genuine help for this task work. Numerous individuals have offered significant remark recommendations on this proposition which gave us the motivation to improve our task. I thank all the individuals for their assistance, straightforwardly and in a roundabout way, in finishing our significant venture.

T Akhilesh Naidu
(2K19/CSE/22)

ABSTRACT

Web one of the modes of availability that is accessible at the doorstep, with admittance to the web one gains admittance to many online stages. An increment in the utilization of these stages gives us a few advantages just as certain downsides. One of such disadvantages is hate speech. Hate speech is a subject of worry for online media stages. With powerfully expanding datasets manual mediation of posts is very inconceivable or will be tedious. Hate speech detection should be an automated task to distinguish hate speech from the provided input. In this paper, we have implemented a deep learning model multi-channel convolutional neural network (MCCNN). The model consists of 3 channels of Convolutional Neural Network. Each channel is merged and connected to a fully connected layer from where the final output is obtained. We have compared our model with a single-channel convolution neural network and results have shown that MCCNN outperformed simple CNN. The accuracy and F1-score achieved by our model are 95.49 and 93.93 for dataset D1 and for dataset D2 97.85% and 95.74% respectively.

CONTENTS

List of Tables

List of Figures

List of Abbreviations and Symbols

1. DL - Deep Learning
2. MCCNN - Multi Channel Convolutional Neural Network
3. CNN - Convolutional Neural Network
4. VM - Virtual Machine
5. NN - Neural Networks
6. NLP - Natural Language Processing

Chapter 1

INTRODUCTION

1.1 A BRIEF OVERVIEW

In the present time and place of the movement of phones and Internet, every individual is related through the network. Digital Media stages give a medium to all people to voice their appraisals and analyze any topic. The power is given to people to impart their insights; notwithstanding the way that it progresses motivation straightforwardly, somebody can essentially notice its malicious effect on the general populace. Various individuals online endeavor to misuse the situation and post basic and devious messages towards people they don't enjoy. These events have out and out influenced the online organization, with various people being engaged by explicit 'harassers,' including acclaimed characters. The targets typically may get affected by these negative comments inciting demoralization and deterioration of enthusiastic health. Due to the sensational improvement of the online organization, Hate Speech is transforming into a basic issue for a tremendous piece of the organization, making the atmosphere noxious and disagreeable for most customers. Scorn talk will spread vibe of hatred between different severe, racial, language or regional social events or stations or organizations. Consequently there is a need to deal with Hate talk in a fast, viable, and versatile way as the human checking of such tremendous proportions of substance is inconceivable. Contempt talk contains a criminal allegation under Section 153A. Many motorized structures to distinguish antagonistic have been proposed using various methodologies, for instance, Bayesian Models, Convolutional Neural Networks, Recurrence Neural Network, Support Vector Machines, etc We used a Multi-Channel Convolutional Neural Network, which will help us to improve results.

1.2 MOTIVATION

The remarkable improvement of online media, for instance, Twitter, Facebook, Instagram, and organization conversations altogether influences our life. It changed correspondence and substance appropriating, but can be used as a phase for disdain talk and the relationship of scorn based activities.

Online scornful talk has a wide scope of impacts as considered from its disengaged accomplice in an unexpected way. The web is available at our doorstep. Along these lines, online stages' affordances are respectably basic and give a colossal group and clear section [17]. Under the Indian Penal Code, it has Sections 153A and 153B; Any showing is a punishable offense that effects or advances disharmony or vibe of hostility or scorn between different severe or racial or phonetic or common social affairs or standings or organizations. In reviews and reports focusing in on adolescents in various territories show rising contempt talk and related bad behaviors reliant on exacting feelings, identity, sexual bearing, or sex, as 80% of respondents have encountered scorn talk on the web and 40% felt attacked or bargained [18]. It offers rise to the essential of brilliant programming models, which thusly perceive all the stunning substance and kill it. As the tremendous size of information on the Web controlling by human noticing isn't essentially possible. It offers climb to the need of sagacious programming models, which subsequently recognize all the stunning substance and kill it.

This issue encouraged us to utilize the force of significant sorting out some way to control bugging by recognizing stunning substance with the objective that the authorities can take fitting actions. So our system will take the comments, Messages, or any statement as data and will mastermind the commitment as toxic or non-hurtful data. Further, the stage can close if to dispose of the substance.

1.3 PROBLEM STATEMENT

This current Project's primary goal is to improve the programmed discovery cycle of Toxic messages and arrange the messages as Hate, Non-Hate and Offensive.
Numerous methodologies have been proposed to recognize disdain discourse, separated into two classes: customary AI techniques, which principally comprise of Support Vector Machine, Logistic Regression and Naïve Bayes, and the subsequent classification is profound learning strategies, which comprise of multi-layer neural organizations to get helpful unique highlights. Customary AI calculations essentially rely upon manual component extractions. An ever increasing number of specialists have tended to loathe discourse discovery issues by utilizing profound learning-based models. We will utilize Multi-Channel Convolution Neural Network to distinguish hate speech using Glove Embedding, an unaided learning strategy to get vector to word portrayal. Multi-Channel Convolution Neural Network (MCCNN) is a convolution Neural Network comprising of numerous CNN layers. MCCNN will assist us with improving outcomes during expectations and will make our model more precise.

Chapter 2

TECHNOLOGY STACK

2.1 TEXT EDITOR -Atom

We have utilized Atom content manager. This Integrated Development Environment (IDE) underpins numerous assorted programming dialects. It has an appended code editorial manager, compiler, terminal, and debugger, which serve both as a machine-level and source-level debugger. It is smoothed out for building and investigating present innovation based web and cloud applications.

2.2 PROGRAMMING LANGUAGE – Python 2.7 and 3.6

The programming language used by us is Python 2.7 and Python 3.6. Python allows efficient integration of systems and an uncomplicated work environment that even a beginner can understand and use quickly and effectively. Since it is an interpreted, general-purpose, and high-level programming language, it can be used to develop networks, servers, and many other applications. It has an object-oriented programming nature. Also, it has a straightforward syntax, which allows the programmers to write exact, logical code efficiently for large and small scale projects.

2.3 GOOGLE COLAB

Google Colab is a cloud administration free of cost, equivalent to Jupyter Notebooks for AI learning and research. It provides an extended platform for runtime of programs completely configured to deep learning and also provides complimentary access to a robust GPU. Using the free Tesla K80 GPU we can implement a wide range of deep learning applications with Google Colaboratory, thereby utilizing Keras, Tensorflow, and PyTorch. The utility of Google Colab for this project

Free GPU support.

1. It provides common access to remote users & developers sharing Jupyter Notebooks and other files, likewise Google docs.
2. Major Python libraries are pre-installed.
3. It is developed on top of Jupyter Notebook.
4. It allows training of DL models free of cost.

2.5 OS VERSION

The operating systems used for our experiments include Windows 10 and Ubuntu 18.04 LTS.

Chapter 3

LITERATURE REVIEW

3.1 Machine and Deep Learning

3.1.1 INTRODUCTION

Social mediums are a platform to express one's voice and should be a place to connect freely without any fear of attack. Hate speech can be defined as speech that expresses one's hatred or stimulates violence towards an individual or group of people based on race, religion, sex, sexual orientation, disease, disability, national origin, gender, color, etc. This definition varies from one country to the other. It includes various forms of expression that spread false information or promote or justify violence, hatred, discrimination against a person or group of persons for several reasons. Usage of offensive and abusive words for people based on any characteristics can harm one's peace and might lead to depression. Therefore, it is necessary to eradicate this problem and demands a tool to rectify it in online platforms. If it remains unaddressed it can lead to serious crimes, violence, and conflicts to a great extent. It's a type of activity that cannot be tolerated and contributes to crime. It sets up a link between violence and hates speech. These events have out and out influenced the online organization, with various people being engaged by explicit 'harassers,' including acclaimed characters. The targets typically may get affected by these negative comments inciting demoralization and deterioration of enthusiastic health Therefore, there must be some tools to detect hate speech on different platforms that are spoiling the social environment. Detection of hate speech and its removal can not only maintain a positive environment in social media but can also prevent the crime rate. It should also be kept in mind while detecting hate speech that freedom of expression is not violated. To explain this, we have few examples of hate speech as below:
Cursing; Considering women as objects; Comments on physical appearance, inferiority; Comparisons, generalization; mocking any events, etc.

Facebook, Google, Microsoft, and Twitter, jointly agreed to a European Union code of conduct obligating them to review the majority of valid notifications for removal of illegal hate speech posted on their services within 24 hours. As we can see the dataset from all the social media platforms will be very large and dynamic, so there is a need to have an automated hate speech detection tool. Many machine learning and deep learning models have shown good results for the detection of hate speech.

In this paper, we have implemented a deep learning model, a multi-channel convolutional neural network. For word embedding, we have used a pre-trained word embedding Glove. This proposed model consists of 3 channels of the convolutional neural network, all the channels are finally merged and connected to the fully connected dense layer from where the output is obtained. We have compared this model to a single-channel simple CNN model and the results have shown that a multi-channel convolutional neural network performed better than simple CNN. We have worked on two different publically available datasets. The two datasets D1, D2 contains 42885 tweets and 24783 tweets respectively and categories the given tweet as hate, offensive and non-hate. Our model achieved an accuracy of 95.49 % and an F1 score of 93.93% while for the second dataset the values are 97.85% and 95.74% respectively.

Machine Learning

It is a subset of AI. It is a powerful technique of training machines so that they can improve from their experience and learn: all this, without being programmed explicitly. The machine can learn from the data accessible in its environment (or experience). This is further used to enhance the overall performance.

Supervised Learning

Supervised learning methods learn from the data in the past and it generalizes for future data. The data in supervised learning is a 'labelled' data. For example, let us say that a training dataset consists of X-Y pairs, (X being the input and Y being the output) where the machine learns an algorithm that evaluates an appropriate output for a sample input. Such a paradigm would need one to have a sample set which represents the machine in observation which can also be used for accuracy assessment of the approach.

Unsupervised Learning

Unsupervised learning is a machine learning technique in which the model works independently to discover patterns and information. The information which is used to train the machine is neither classified nor labeled. It's the responsibility of the machine to group the data according to patterns, similarities, and differences without any prior training.

The goal of this kind of learning is to model the structure to learn more about data. Unsupervised learning is further classified into two types Association and Clustering.

Deep Learning

Deep Learning is a subpart of ML which consists of an ANN. A NN represents the working of the human brain. The structure and functioning of the brain inspire an Artificial Neural Network. Deep Learning AI can learn even if the data is unlabeled or unstructured. Deep Learning is commonly utilized for identifying objects, perceiving discourse, deciphering dialects, and deciding.

Supervised Deep Learning

A machine can itself figure out the parameters required to detect or classify tasks solely based on raw-data representation learning. Unlike this, other conventional machine learning techniques are not able to deal with sample data in the raw form. Because of multiple levels of representation, the change in model from a low level to a higher abstract one is achieved. Sufficient quantities of these transformations have enabled machines to learn more complex functions. The deep learning techniques have also shown a better performance than the conventional algorithms for various machine learning problems, some of which include recognition of speech, object detection, intrusion detection, and NLP understanding.

The DL models are divided into 3 groups:

- Hybrid models use both discriminative and generative models.
- Generative models employ unsupervised learning strategies
- Supervised learning methods are used by Discriminative models.

3.1.2 Neural Network

Neural Networks are AI calculations that work comparably to the human sensory system. A NN is an organization or circuit of neurons. Neural organization (NN), on account of fake neurons called counterfeit(Artificial) neural Network (ANN) .In more commonsense terms, neural organizations are non-direct measurable information demonstrating or dynamic apparatuses. They can be utilized to demonstrate complex connections among sources of info and yields or to discover designs in information.

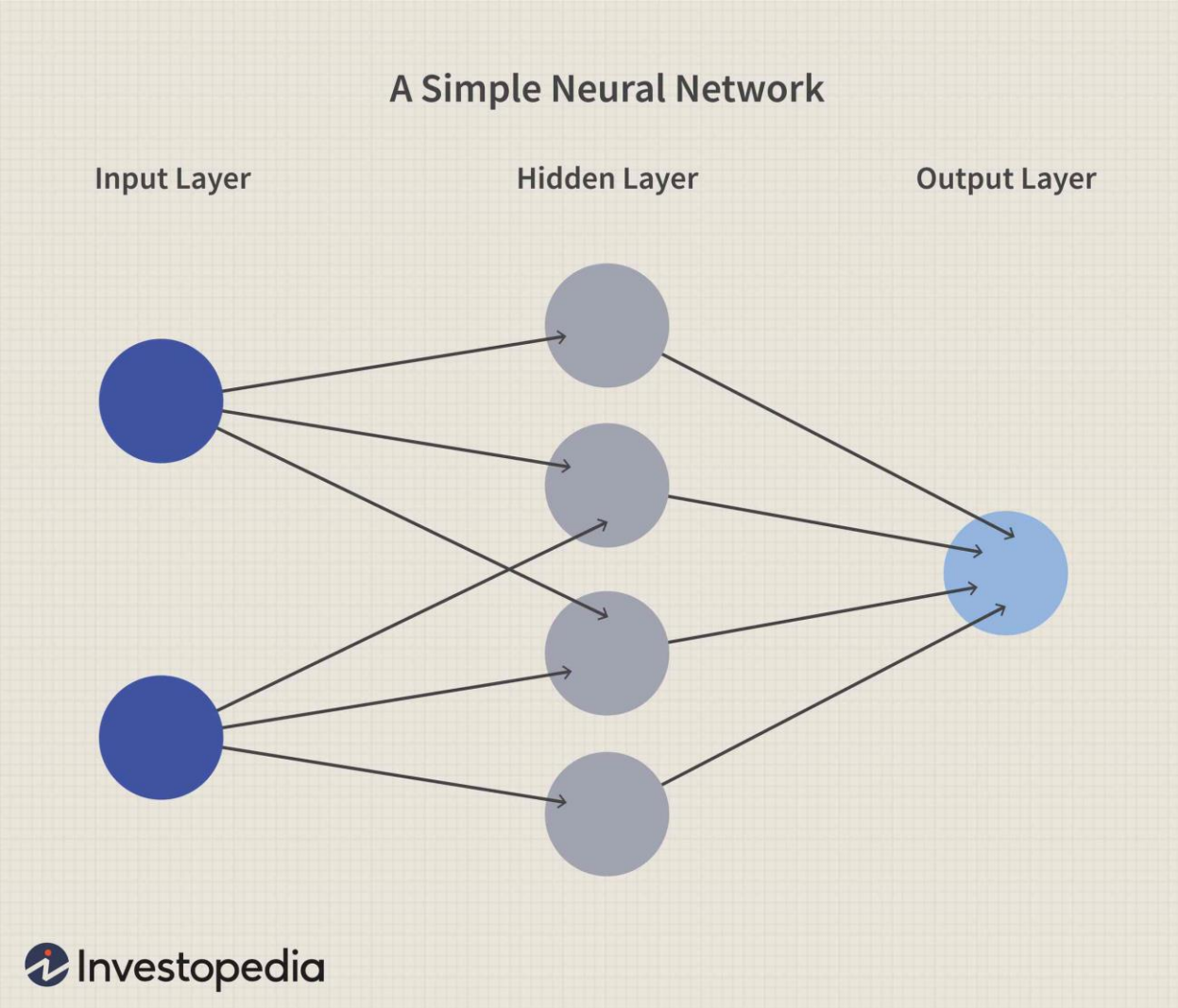


Figure 3.1 – A simple NN

3.1.3 Convolutional Neural Network

CNN is one of the most mainstream Deep Neural Network. CNN has a fantastic presentation in AI issues. CNN comprises of numerous layers, including convolutional layer, non-linearity layer, pooling layer, and fully-connected layer [3]. The convolutional and fully-connected layers have boundaries, however pooling and non-linearity layers don't have boundaries. The aftereffects of CNN are momentous in the previous few decades. CNN is on the upper side since it decreases the boundary in ANN. CNN is generally utilized in situations where the highlights are not spatially needy.

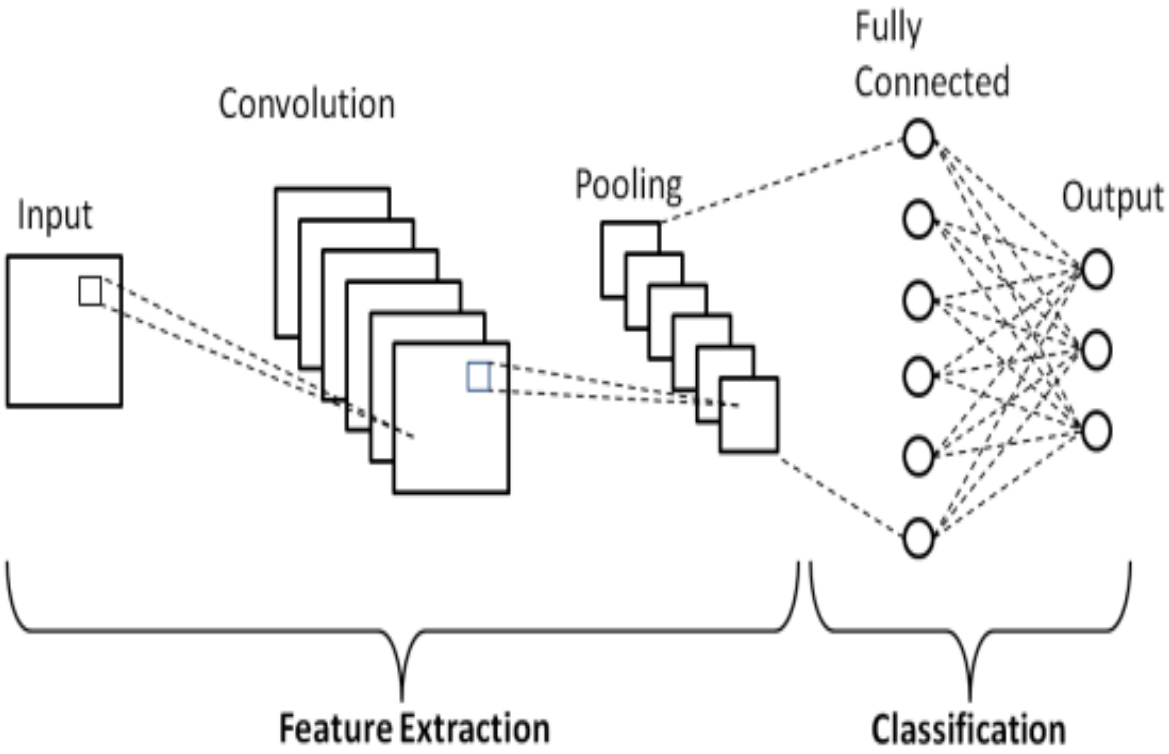


Figure 3.2 - CNN

3.1.4 Multi-Channel Convolutional Neural Network

We have proposed a multi-channel architecture to improve the efficiency of the existing basic model. This will improve the tuning of the filters. For our model, we have implemented a 3-channel convolutional neural network [15]. Each channel contains an input layer, embedding layer, the same number of convolutional layers, a max-pooling layer, a flattening layer. The parameter used for all the layers is the same with the same weights. The layers are merged and connected to a fully connected layer from which the output is obtained. The Convolutional layer consists of 32 filters with a kernel size of 5 and activation function ReLU. The fully connected dense layer will produce a 32 unit output. Fig-3 shows the MCCNN model used in this paper.

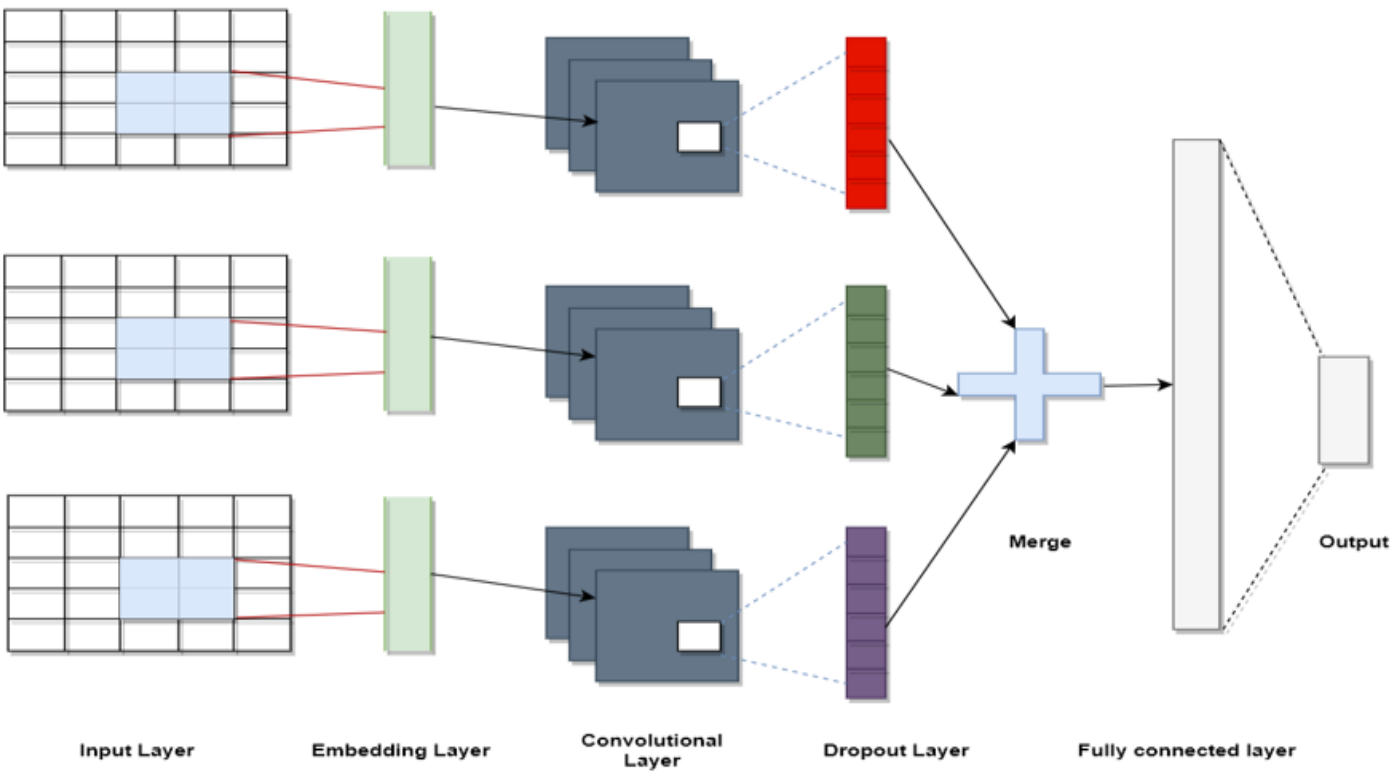


Fig 3:- Multi Channel Convolutional Neural Network

1. **Input Layer**

In this layer, every one of the groupings is changed over to a number structure where every token has been relegated to a special file. The information groupings are then zero-cushioned to have an equivalent length as it helps in improving execution by keeping data saved at the lines.

2. **Embedding Layer**

We have used Glove which is a pre-trained word embedding model. We have already explained the working of Glove in the previous section. In this layer, a word-word co-occurrence matrix is created where a row represents a word and a column represents a context. The global matrix factorization method is used on this occurrence matrix created. The final result of glove embedding is a word to a vector representation. These vectors are given as input to our deep learning model used in this paper. We have used global_text which is globally available.

3**. Convolutional Layer**

The convolutional layer is the integral unit of a CNN where by far, most of the estimation is incorporated. Convolutional layer is the first layer from which input is passed. It consist of feature maps with neurons organized in them. The limits of the layer are a lot of learnable channels or pieces. Neurons that lie in a similar component map share the weight (limit sharing), thus lessening the association's unpredictability by keeping the number of limits low. The spatial loosen up of inadequate network between the neurons of two layers is a hyperparameter called open field. The hyperparameters that control the yield volume's size are the depth, stride, and zero paddings.

4**. Pooling Layer**

Principal CNN configuration have trading pooling layers and conv layers and the last abilities to reduce the estimation of the order maps (No loss of information) likewise, the amount of limits in the net and hence diminishing the for the most part computational eccentricity. The problem of overfitting is solved here. Some examples of pooling are max pooling, typical pooling, stochastic pooling, spooky pooling, spatial pyramid pooling. Fig 3.4 shows the movement of max pooling.

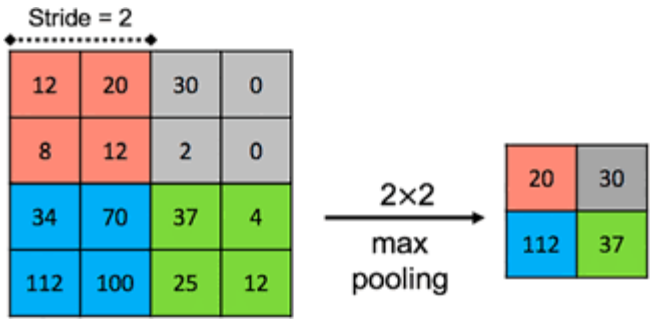


Figure 3.4 Max Pooling

5. **Fully Connected Layer**

The fully connected layer is the last layer of the NN. The Neurons in this layer are associated with all neurons in the previous layers, as in an ordinary NN. As Neurons play a significant role, this layer is responsible for thinking. The neurons are not one-dimensional. Some designs have FCL after the pooling layer.

II. Actuation FUNCTIONS

Activation Functions are numerical conditions that decide the yield of a NN. The capacity is connected to every neuron in the organization, and decides if it should be enacted ("terminated") or not, founded on whether every neuron's info is important for the model's forecast. Enactment works additionally help standardize the yield of every neuron to a reach somewhere in the range of 1 and 0 or between - 1 and 1.

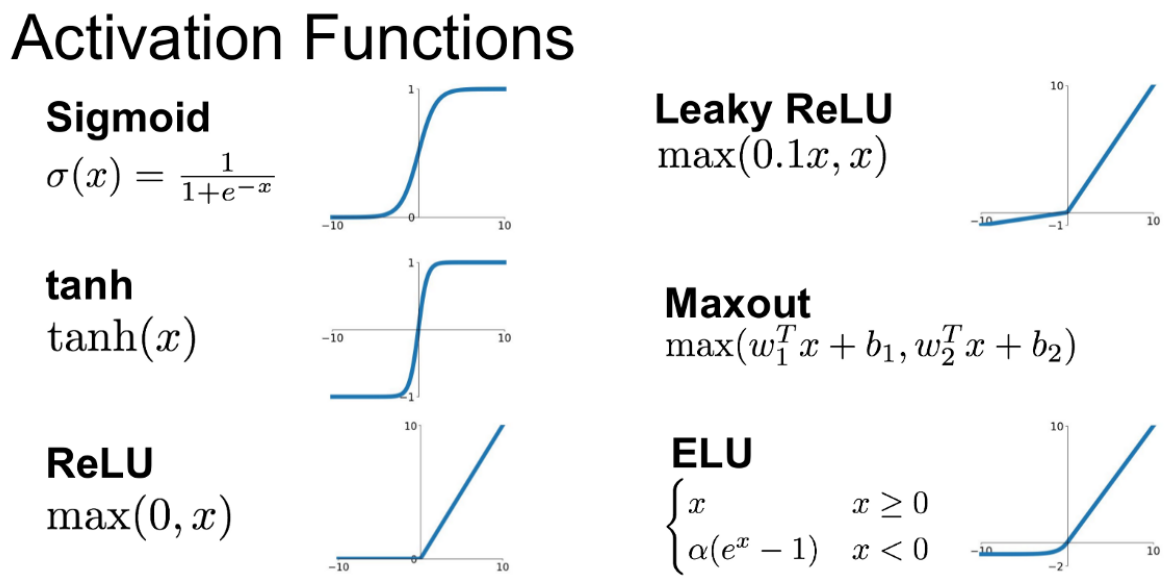


Figure 3.5- Activation Function

3.2 Word Embeddings

A word embedding is a technique or algorithm in which words having similar meaning or words having some relations are kept together by assigning vectors values that are nearly the same.

This approach to manage addressing words and reports may be seen as one of the fundamental forward jumps of significant learning on testing trademark language planning issues.

Word embeddings are a class of strategies where individual words are addressed as real regarded vectors in a predefined vector space. Each word is intended to one vector, and the vector regards are discovered in a way that resembles a neural association, and subsequently, the method is consistently lumped into the field of significant learning. The path into the strategy is using a thickly scattered depiction for each word. An authentic regarded vector addresses each word, routinely tens or a few estimations. This is separated from the huge numbers or numerous sizes required for pitiful word depictions, for example, a one-hot encoding.

Word2Vec [4]

Word2Vec is a word embedding technique created by Tomas Mikolov at Google in 2013. It is a solid strategy for productively taking in an independent word installing from a book corpus. It

will create a vector for word and have a linear distribution between words with similar meanings or closely related words

The work also included an investigation of the educated vectors and the investigation of vector math on the portrayals of words. For instance, that taking away the "manness" from "Lord" and adding "ladies ness" brings about "Sovereign," catching the 22 relationships "the ruler is to the sovereign as man is to a lady."

3.2.1 Glove Embedding

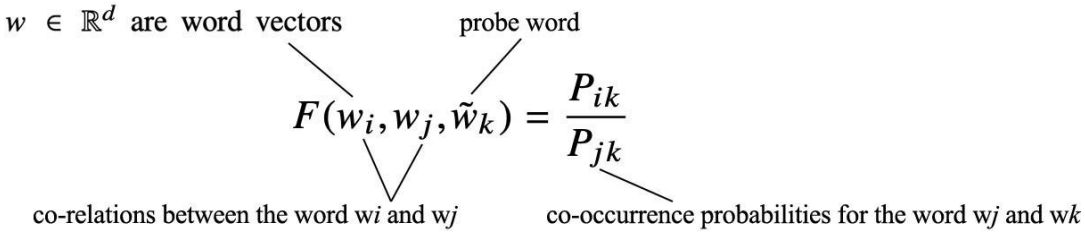
GloVe which is short term used for global vectorization for words installing However, it utilizes an alternate component and conditions to make the installing grid. To examine GloVe, how about we characterize the given statements.

X_{ij} tabulate the number of times word j occurs in the context of word i.

$X_i = E_k X_{ik}$

$P_{ij} = P_{(j|i)} = X_{ij}/X;$

Co-event probabilities Proportion is shown below:



This extent gives us some comprehension on the co-association of the test word W_k with the word w_i and w_j.

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Very small or large:
solid is related to ice but not steam, or
gas is related to steam but not ice

close to 1:
water is highly related to ice and steam, or
fashion is not related to ice or steam.

The extent can be pretty much nothing, tremendous or identical to 1 depends upon their connections. For example, if the volume is colossal, the test word is related to w_i yet not w_j. Hints can be found on the relations between three extraordinary stories. Naturally, we can relate it to 3-bigram or bigram.

We need to develop a model for F given some appealing behaviour we need for the introducing vector w. As discussed already, linearity is critical in the word introducing thought. So if a system is set up on this standard, we should expect that F can be reformulated a

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{P_{ik}}{P_{jk}}$$

($w_i{}^T \tilde{w}_k$ relate to (high probability if they are similar); $w_j{}^T \tilde{w}_k$)

Where, we simply need to figure the distinction and the similitude of word inserting for the boundaries in F. Furthermore, to implement such balance, we can have

$$F\left((w_i - w_j)^T \tilde{w}_k\right) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}$$

Instinctively, we are keeping up the direct relationship among all these installing vec

$$F(w_i^T \tilde{w}_k) = P_{ik} = \frac{X_{ik}}{X_i}$$

Since $F(x) = \exp(x)$,

$$w_i^T \tilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$$

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log(X_{ik})$$

(X_{ik}: co-occurrence count for word w*i* and w*k*)

We can acclimatize log (X_i) as a reliable tendency term since it is invariant of k. In any case, to keep up the even need among I and k, we will part it into two tendency terms above. This w and b structure the introducing grid. Subsequently, the spot aftereffect of two embedding networks predicts the log co-occasion check.

Glove which is developed by Stanford, is a method for creating word embedding. In this method, we find out co-occurrence between words and contexts, and this is how a word-word co-occurrence matrix is created where a row represents a word and a column represents a context. The global matrix factorization method is used on this occurrence matrix created. The final result of glove embedding is a word to a vector representation. These vectors are given as input to our deep learning models used in this paper.

Chapter 4

RELATED WORK

In this segment, we present the phrasing of scorn discourse and the standards of cutting edge profound learning strategies. Finding hate speech is the subject of an Investigative conversation. Many researchers are working in this field for automated hate speech detection in various platforms. Various models have been proposed for the detection of hate speech. Many researchers had worked on various standard machine learning models as well as deep learning models. Studies have shown that deep learning models have shown tremendous results for the detection of hate speech. Starting with some traditional machine learning models, [1] used SVM classifiers for hate speech detection and got some good results. [2], used a naïve base classifier for the classification of tweets. [3], used logistic regression (LR) for detection and achieved a precession of 90%. , [4] focused on two words racism and sexism: to detect hate speech on Twitter. [5], prepared a multi-class classifier to recognize offensive and hate language in tweets. [6], have used Recurrence Neural Network with uni-gram and bi-gram character embedding for hate speech detection. [7], used word2vec word vectors and CNN as a classifier which gave some good results. [8], proposed a deep neural network for hate speech detection and achieved an F1-score of 92%. [9] used CNN-GRU model for hate speech detection on the public dataset, as well as a private dataset, and the model, performed well on all the datasets.[10] used RNN for the detection of hate speech and to distinguish between racism and sexism.[14] proposed a Multi-channel CNN and Bi-direction GRU-based model for text sentiment analysis and achieved an accuracy of 91.20% for the IMDB dataset.[11] used Bi-directional LSTM with word2vec and achieved an accuracy of 91.10%. [12] used unigram and SVM classifier for two different datasets, the first one for binary classification as hate and non-hate, the second dataset for ternary classification, detecting tweets as hate, offensive or clean. The model achieved an accuracy of 87.4 % for the first dataset and an accuracy of 78.4 for the second dataset.[13] detected hate speech for multiple languages and datasets using multi-channel BERT and got some good results.[15] proposed a fusion-based model in which they did a fusion of three convolutional neural networks and achieved a mean accuracy of 75.4% and F1-Score of 70.4%.[16] used multi-nominal logistic regression for hate speech detection on social media Twitter and achieved an accuracy of 87.04%. In this paper we have used multiple channels of CNN, each channel has similar parameters with similar weight and filters. We got some good results in terms of detecting hate speech in two different datasets.

Chapter 5

PROPOSED WORK

We proposed to implement Multi-Channel Convolutional Neural Network to detect hate speech automatically. It will give an output as to whether the comment is Hate, Non-Hate or offensive. For word embedding, we used Glove Embedding, which will convert words to vector.

5.1 Dataset and its Structure

In our Deep Convolutional Neural Network model, the dataset that we used consist of 164354 comments tagged as toxic or non-toxic. The comments that are there in the dataset are collected from different resources.

	class	tweet
13932	1	Pumpkin spice Marlboro's for da hoes
1636	1	“@_CiaraaaS: What things do you love? &#...
23100	2	Yankees should have NEVER gave away Melky. Guy...
3498	1	@Im_Thirst I love Louis CK! Quit bein a faggot...
12999	1	My boyfriend is such a smart ass bitch watch y...
195	1	"@Montrell_: I'm tired of bitches saying I loo...
1916	1	“@talofabreeze: @uce_INA @snoleezey801 D...
7860	1	Awwwwww RT @Tyga: That ain't my hoe.
2011	2	<-- nudes are trash
17806	1	RT @TooMessedUp: Fucking retarded question... ...

Table 1:- Dataset1 Structure

	Index	Tweet	Class
42793	42794	@TweetsAndFreaks I told u ugly niggas get bitc...	1
33634	33635	school me ya kahi vandemataram bolne pe pure B...	2
14155	14156	and if that's yo bitch then you may have to fi...	1
33368	33369	At work I was forced to listen to the most dis...	2
15752	15753	@freshestcrook @_MamaTosha lmao he lyin sis! H...	1
4468	4469	“@NoRapist: Throwing lamps at bitches wh...	1
16022	16023	Woke up the next morning like "why in THEE FUC...	1
36585	36586	RT @Persianboi10: #YesAllMen these hoes ain't ...	1
20687	20688	@aubsceneone it's jolly trash with wack ass bells	1
34304	34305	Cavs trash too	2

Table 2:- Dataset2 Structure

5.2 Word Embedding

We used Glove Embedding which will give us the word2vec representation. For that, we have used Tenserflow library from which we imported Embedding. The code for Glove Embedding is shown below. It consist of 40000 word vectors in Glove. We have used a public available glove text file.

```
MAX_SEQUENCE_LENGTH = 1000
embeddings_index = {}
f = open('glove.6B.100d.txt', encoding="utf8")
for line in f:
    values = line.split()
    word = values[0]
    coefs = np.asarray(values[1:], dtype='float32')
    embeddings_index[word] = coefs
f.close()

print('Total %s word vectors in Glove.' % len(embeddings_index))

embedding_matrix = np.random.random((len(word_index) + 1, 100))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix[i] = embedding_vector

embedding_layer = Embedding(len(word_index) + 1, 100, weights=[embedding_matrix], input_length=1000)
```

```
Total 400000 word vectors in Glove.
```

Figure 5.1 Glove Embedding code

5.3 Convolutional Neural Network

The model we used here is the CNN model for text characterization. We have used Glove Embedding for initial Pre-Processing. Fig-5.2 Shows the CNN model used, it contains the following layers, starting with the input layer (embedding layer) followed by a dropout layer which is connected to a convolution layer followed by a pooling layer, a hidden dense layer. There are multiple parallel convolutional layers with some kernel size, and the activation function ReLU. The convolutional layer extracts the features so that data can be presented in a better way. Padding is added to maintain the input and output length. Pooling layer: The function of layer integration is gradually reduced the area of representation to reduce the number of parameters and computation in the network and apply to the map of each item (channels) independently, the output from the Pooling layer is connected as an input to a dropout layer to solve overfitting problem. Lastly, a fully connected dense layer with the ReLU triggers to generate the final prediction.

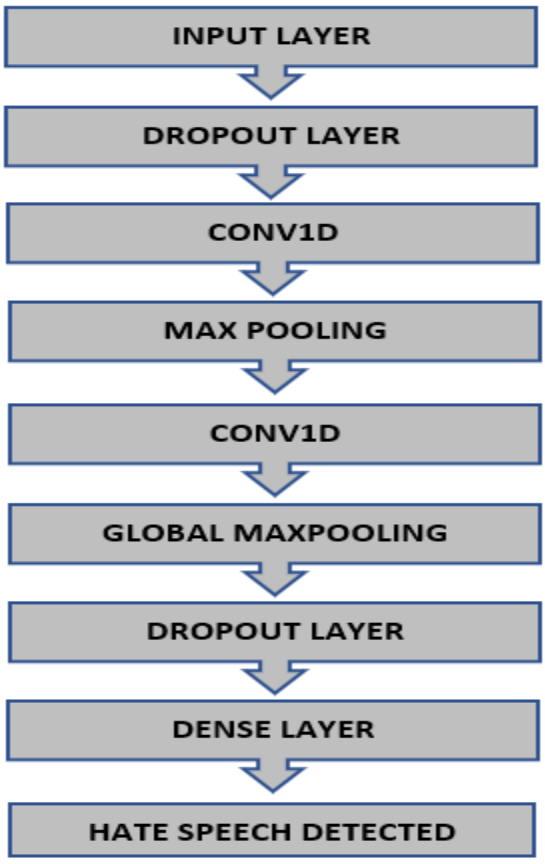


Fig 5.2 CNN

5.3 Multi-Channel Convolutional Neural Network

Many libraries are available on the internet that can be used, out of which Google's TensorFlow is in high demand. It is a freely available programming library mostly used for high computational occupations utilizing information stream charts where edges signify tensors. With this, a singular API can be used to course stack between various center points (CPUs or GPUs). This library is transparently available since November 2015. Keras is seen as the second, rapidly creating a robust learning framework. This open-source library written in Python is suitable for running on top of TensorFlow or Theano. Theano is an open-source Python library for numerical counts and unravels the pattern of making significant learning models.

Layer (type)	Output Shape	Param #	Connected to
input_4 (InputLayer)	[(None, 1000)]	0	
input_5 (InputLayer)	[(None, 1000)]	0	
input_6 (InputLayer)	[(None, 1000)]	0	
embedding_1 (Embedding)	(None, 1000, 100)	3637700	input_4[0][0] input_5[0][0] input_6[0][0]
conv1d_3 (Conv1D)	(None, 996, 32)	16032	embedding_1[0][0]
conv1d_4 (Conv1D)	(None, 996, 32)	16032	embedding_1[1][0]
conv1d_5 (Conv1D)	(None, 996, 32)	16032	embedding_1[2][0]
max_pooling1d_3 (MaxPooling1D)	(None, 199, 32)	0	conv1d_3[0][0]
max_pooling1d_4 (MaxPooling1D)	(None, 199, 32)	0	conv1d_4[0][0]
max_pooling1d_5 (MaxPooling1D)	(None, 28, 32)	0	conv1d_5[0][0]
flatten_3 (Flatten)	(None, 6368)	0	max_pooling1d_3[0][0]
flatten_4 (Flatten)	(None, 6368)	0	max_pooling1d_4[0][0]
flatten_5 (Flatten)	(None, 896)	0	max_pooling1d_5[0][0]
concatenate_1 (Concatenate)	(None, 13632)	0	flatten_3[0][0] flatten_4[0][0] flatten_5[0][0]
dense_2 (Dense)	(None, 32)	436256	concatenate_1[0][0]

Figure 5.3 MCCNN model

Chapter 6

Experiments & Results

6.1 Experimental Results

In our proposed work, we have used Multi-Channel Convolutional Neural Network (MCCNN) for the detection of hate speech. We have worked on two different datasets. Dataset D1 consists of 42885 tweets which are categorized as Hate, Offensive, and Non-hate. The second dataset D2 consists of 24783 tweets categorized as Hate, Offensive, and Non-Hate. We have compared our model with simple CNN and LSTM models. We have done the performance evaluation based on the Precession, Recall, F1 score and, accuracy. Precession will tell us about how the model performed for the actual positive class. The recall will tell us how well our model performed for overall positive examples. F1-score will give a balance value for precession and recall. Accuracy will tell us how precise our model is. The results obtained by running our model are shown in Table 1. The model achieved an accuracy of 95.49% and an F1-score of 93.93% for dataset D1 and for dataset D2 the values are 97.85% and 95.74%.

```
_________________________________________________________________
Epoch 1/10
620/620 [==============================] - 256s 410ms/step - loss: 0.3574 - accuracy: 0.7967 - val_loss: 0.2540 - val_accuracy: 0.8534
Epoch 2/10
620/620 [==============================] - 254s 410ms/step - loss: 0.2364 - accuracy: 0.8659 - val_loss: 0.2176 - val_accuracy: 0.8752
Epoch 3/10
620/620 [==============================] - 254s 409ms/step - loss: 0.1912 - accuracy: 0.8932 - val_loss: 0.2013 - val_accuracy: 0.8857
Epoch 4/10
620/620 [==============================] - 255s 411ms/step - loss: 0.1563 - accuracy: 0.9149 - val_loss: 0.1991 - val_accuracy: 0.8874
Epoch 5/10
620/620 [==============================] - 255s 411ms/step - loss: 0.1275 - accuracy: 0.9310 - val_loss: 0.1998 - val_accuracy: 0.9018
Epoch 6/10
620/620 [==============================] - 256s 412ms/step - loss: 0.0965 - accuracy: 0.9494 - val_loss: 0.2003 - val_accuracy: 0.9008
Epoch 7/10
620/620 [==============================] - 254s 410ms/step - loss: 0.0728 - accuracy: 0.9618 - val_loss: 0.2211 - val_accuracy: 0.9060
Epoch 8/10
620/620 [==============================] - 261s 422ms/step - loss: 0.0599 - accuracy: 0.9680 - val_loss: 0.2675 - val_accuracy: 0.9089
Epoch 9/10
620/620 [==============================] - 262s 422ms/step - loss: 0.0500 - accuracy: 0.9738 - val_loss: 0.2779 - val_accuracy: 0.9103
Epoch 10/10
620/620 [==============================] - 261s 420ms/step - loss: 0.0451 - accuracy: 0.9747 - val_loss: 0.2894 - val_accuracy: 0.9035
```

Figure 6.1 Epoch plots for Dataset1

```
_________________________________________________________________
Epoch 1/10
1097/1097 [==============================] - 557s 507ms/step - loss: 0.3803 - accuracy: 0.7397 - val_loss: 0.2872 - val_accuracy: 0.8129
Epoch 2/10
1097/1097 [==============================] - 556s 507ms/step - loss: 0.2233 - accuracy: 0.8558 - val_loss: 0.2769 - val_accuracy: 0.8310
Epoch 3/10
1097/1097 [==============================] - 552s 503ms/step - loss: 0.1717 - accuracy: 0.8937 - val_loss: 0.2881 - val_accuracy: 0.8400
Epoch 4/10
1097/1097 [==============================] - 562s 512ms/step - loss: 0.1352 - accuracy: 0.9172 - val_loss: 0.3192 - val_accuracy: 0.8466
Epoch 5/10
1097/1097 [==============================] - 557s 508ms/step - loss: 0.1069 - accuracy: 0.9366 - val_loss: 0.3588 - val_accuracy: 0.8499
Epoch 6/10
1097/1097 [==============================] - 555s 506ms/step - loss: 0.0933 - accuracy: 0.9430 - val_loss: 0.3812 - val_accuracy: 0.8461
Epoch 7/10
1097/1097 [==============================] - 553s 504ms/step - loss: 0.0810 - accuracy: 0.9528 - val_loss: 0.4264 - val_accuracy: 0.8485
Epoch 8/10
1097/1097 [==============================] - 555s 506ms/step - loss: 0.0743 - accuracy: 0.9553 - val_loss: 0.4319 - val_accuracy: 0.8506
Epoch 9/10
1097/1097 [==============================] - 554s 505ms/step - loss: 0.0671 - accuracy: 0.9587 - val_loss: 0.4557 - val_accuracy: 0.8493
Epoch 10/10
1097/1097 [==============================] - 552s 503ms/step - loss: 0.0684 - accuracy: 0.9595 - val_loss: 0.4825 - val_accuracy: 0.8523

Model Training Completed !
```

Fig 6.2: Epoch Plot for Dataset2

6.2 Summary of Results Obtained

Dataset	Model	Precession	Recall	F1-Score	Accuracy
Dataset D1	**CNN**	**89.14**	**85.88**	**87.46**	**90.24**
	MCCNN	**95.43**	**92.59**	**93.93**	**95.49**
Dataset D2	**CNN**	**95.14**	**84.43**	**88.64**	**95.89**
	MCCNN	**96.54**	**93.65**	**95.74**	**97.85**

Table 3: Result Obtained

Graph Plot for the comparison of CNN and MCCNN

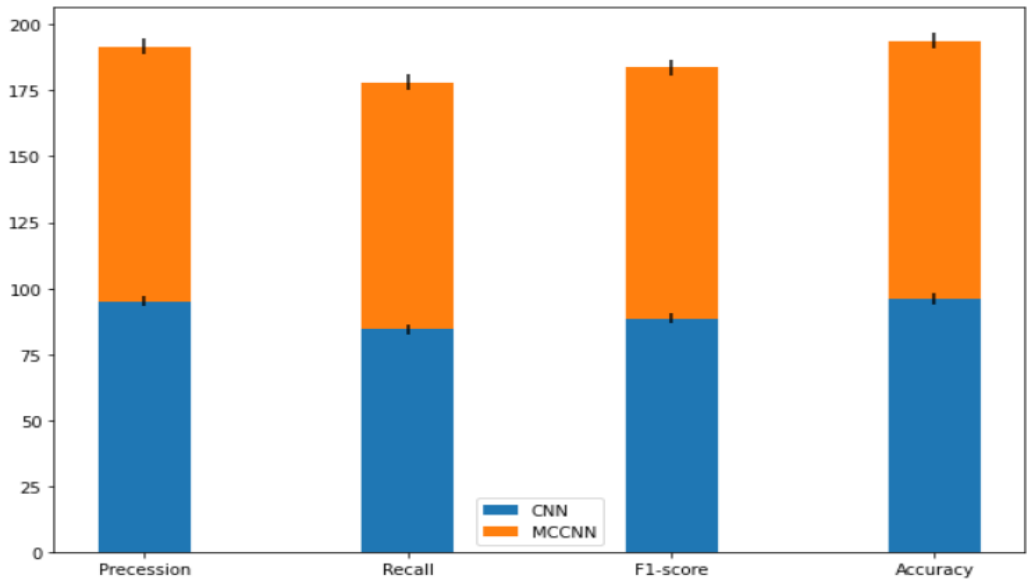


Fig 6.3 Plot for Dataset D3

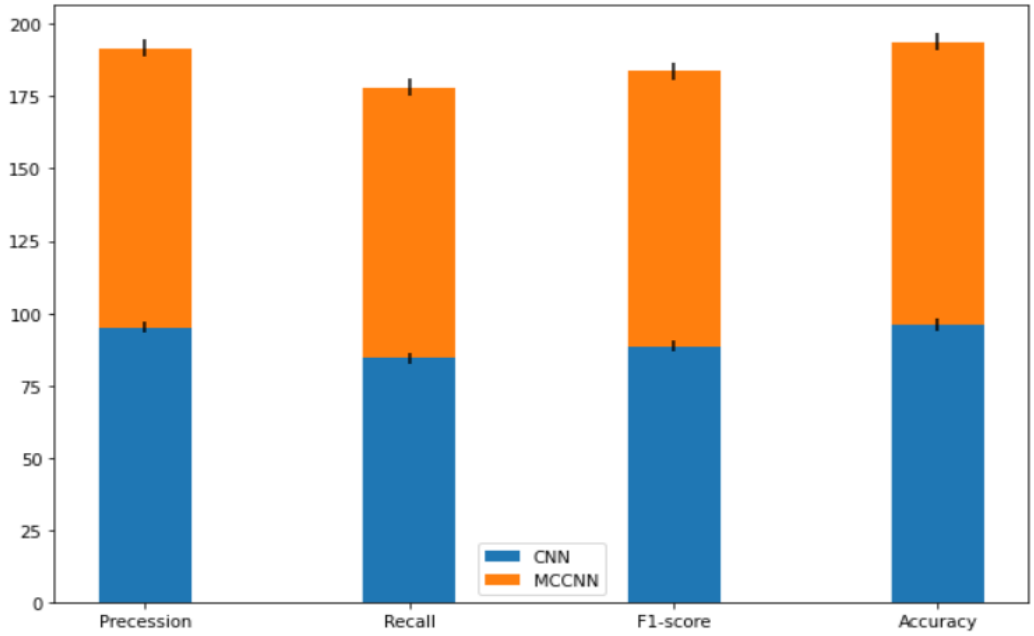


Fig 6.4 Plot for dataset D2

Chapter 7

Conclusion and Future Work

Multi-Channel Convolutional Neural Network has been effective over various issue areas, and the equivalent can is found on account of Hate Speech Detection, as this report illustrates. Parallel arrangement of remarks is an invaluable way to deal with banner harmful and non-poisonous remarks. As NLP acts various types of difficulties like the language can't be seen totally dependent on phonetic guidelines, NLP can't generally be dealt with utilizing local directed learning calculations. MCCNN follows an alternate way to deal with getting language. It is a fairly human-like way to deal with comprehend language through experimentation. Joined with Deep Learning, support learning turns out to be much more remarkable and fit for understanding characteristic language and give more promising results.

In our proposed work we have implemented a Multi-channel convolutional neural network for hate speech detection in various platforms. For our model, we have used the pre-trained word embedding model Glove (Global Vector for word representation). We have compared the result of the MCCNN model with the simple CNN model. It can be seen from the obtained result that MCCNN performed better than simple models. Our model achieved an accuracy of 95.49% and an F1-score of 93.93 for dataset D1 and for dataset D2 the values are 97.85 and 95.74.

Short Description of Work done in Paper 1:-

Abstract— The Internet one of the mediums of connectivity that is available at the doorstep, with access to the internet one gets access to many web-based platforms. An increase in the use of these platforms gives us some benefits as well as some drawbacks. One of such drawbacks is hate speech. Hate speech is a topic of concern for social media platforms. With dynamically increasing datasets manual intervention of posts is quite impossible or will be time-consuming. Hate speech detection is an automated task to detect hate speech from the input. In this paper, we have compared some deep learning models like Convolution Neural Network (CNN), Recurrence Neural Network (RNN), Long Gated Recurrent Unit (GRU), and Long-Short Term Memory. The datasets used here are publicly available. The result of our analysis shows us that GRU performed better than other basic deep learning models. The model achieved an accuracy of 92.60% with an F1 score of 81.84% for dataset (D1) and for dataset (D2) the accuracy and F1 score is 96.15 and 83.06 respectively.

Datasets: We have used two different datasets. Both the datasets are public available datasets. Dataset (D1) has 2,23,549 comment texts labeled as toxic and non-toxic. Dataset (D2) has 29,530 collections of tweets from Twitter labeled as toxic and non-toxic

Result and Discussion

In our proposed work we have compared some deep learning models for the detection of hate speech on two different datasets D1 and D2. Both the datasets are publicly available. D1 dataset contains 2,23,549 comment text out of which 159571 are used to train the model. D2 dataset contains 29530 collections of tweets from Twitter. We have done the performance evaluation based on the Precession, Recall, F1 score and, accuracy. Precession will tell us about how the model performed for the actual positive class. The recall will tell us how well our model performed for overall positive examples. F1-score will give a balance value for precession and recall. Accuracy will tell us how precise our model is. Table 1 shows the results obtained after running the models, CNN, RNN, LSTM, and GRU. The result shows that all the models showed good results, GRU got the highest F1 score and accuracy of 81.84% and 92.60% respectively for dataset (D1). Fig 5 shows a comparison of all the models used in this paper for dataset (D1).

Table 8.1 Result Obtained for Dataset1

Models	Dataset(D1)			
	Precession	Re-call	F1-Score	Accuracy
CNN	76.74	86.96	80.67	92.03
RNN	75.22	85.60	78.98	91.31
LSTM	77.47	88.46	81.65	92.38
GRU	**77.94**	**87.87**	**81.84**	**92.60**

Models	Dataset(D2)			
	Precession	Re-call	F1-Score	Accuracy
CNN	84.24	78.25	80.92	95.54
RNN	86.39	74.91	79.97	95.71
LSTM	87.13	77.39	81.41	95.86
GRU	**87.15**	**79.95**	**83.06**	**96.15**

Table 8.2 Result obtained for dataset D2

Graph plot for above dataset

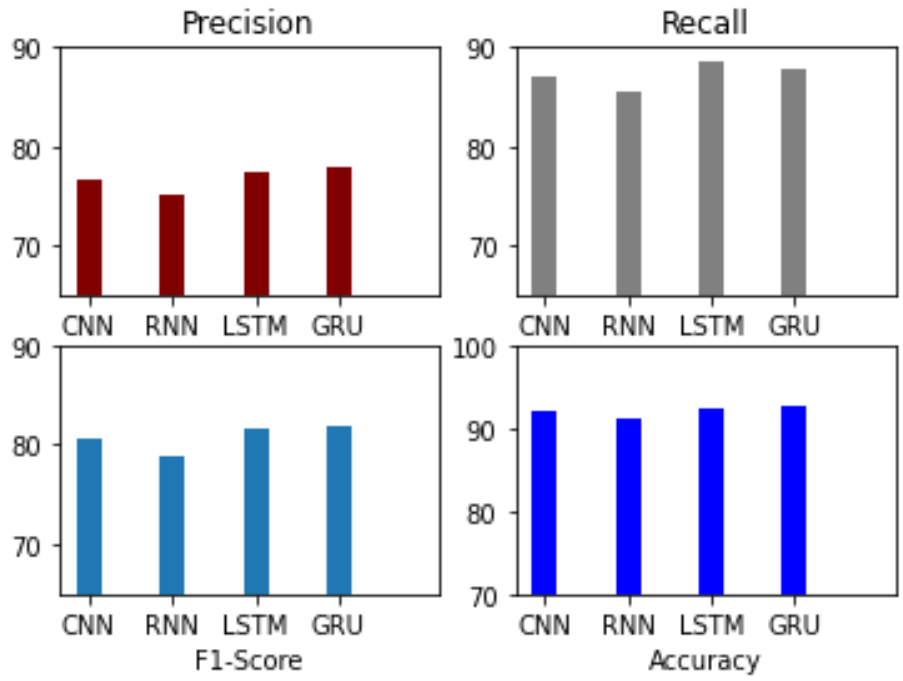


Fig 8.3 Graph plot for dataset D1

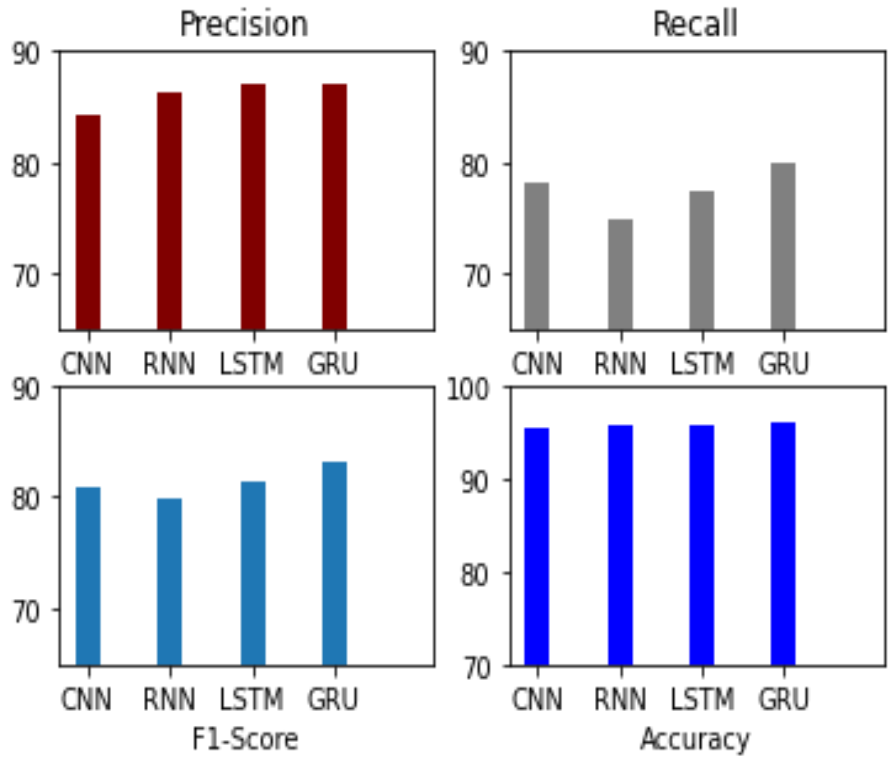


Fig 8.4 Graph plot for dataset D2

Conclusion

In this paper, we have looked at some deep learning models for hate speech detection. For word embedding, we have utilized a pre-trained model Glove Embedding. These Pre-Trained models will give us a word to vector portrayal, which was given as contribution to our DNN models, CNN, RNN, LSTM, and GRU. The results showed that GRU performed best for both datasets with F1 scores 81.84, 83.06, and accuracy 92.60, 96.15 respectively. In the future, we can try these models on the multilingual dataset.

REFERENCES

[1] T. Joachims, ‘‘Making large-scale SVM learning practical,’’Komplex itätsreduktion Multivariaten Datenstrukturen, TU Dortmund, Dortmund, Germany, Tech. Rep. 28, 1998.

[2] I. Kwok and Y. Wang, ‘‘Locate the hate: Detecting tweets against blacks,’’ in Proc. 27th AAAI Conf. Artif. Intell., 2013, pp. 1621–1622

[3] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. arXiv preprint arXiv: 1703.04009, 2017.

[4] Waseem, Z.; Hovy, D. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In Proceedings of the NAACL Student Research Workshop, San Diego, CA, USA, 12–17 June 2016; Association for Computational Linguistics: San Diego, CA, USA, 2016; pp. 88–93.

[5] Davidson, T.; Warmsley, D.; Macy, M.; Weber, I. Automated Hate Speech Detection and the Problem of Offensive Language. arXiv 2017, arXiv:1703.04009.

[6] Yashar Mehdad and Joel Tetreault. 2016. Do characters abuse more than words?. In Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue. 299–303

[7] P. K. Roy, A. K. Tripathy, T. K. Das and X. -Z. Gao, "A Framework for Hate Speech Detection Using Deep Convolutional Neural Network," in IEEE Access, vol. 8, pp. 204951-204962, 2020, doi: 10.1109/ACCESS.2020.3037073.

[8] Zhang, Z., Robinson, D., Tepper, J.: Detecting hate speech on twitter using a convolution-gru based deep neural network. In: The Semantic Web, pp. 745–760. Springer International Publishing, Cham (2018)

[9] Zhang, Z., Robinson, D., Tepper, J.: Detecting hate speech on twitter using a convolution-gru based deep neural network. In: The Semantic Web, pp. 745–760. Springer International Publishing, Cham (2018)

[10] Pitsilis, G.K.; Ramampiaro, H.; Langseth, H. Effective hate-speech detection in Twitter data using recurrent neural networks. Appl. Intell. 2018, 48, 4730–4742.

[11] Isnain, Auliya Rahman, Agus Sihabuddin, and Yohanes Suyanto. "Bidirectional Long Short Term Memory Method and Word2vec Extraction Approach for Hate Speech Detection." *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)* 14.2 (2020): 169-178.

[12] H. Watanabe, M. Bouazizi and T. Ohtsuki, "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection," in IEEE Access, vol. 6, pp. 13825-13835, 2018, doi: 10.1109/ACCESS.2018.2806394.

[13] H. Sohn and H. Lee, "MC-BERT4HATE: Hate Speech Detection using Multi-channel BERT for Different Languages and Translations," *2019 International Conference on Data Mining Workshops (ICDMW)*, 2019, pp. 551-559, doi: 10.1109/ICDMW.2019.00084.

[14] Y. Cheng, L. Yao, G. Xiang, G. Zhang, T. Tang and L. Zhong, "Text Sentiment Orientation Analysis Based on Multi-Channel CNN and Bidirectional GRU With Attention

Mechanism," in IEEE Access, vol. 8, pp. 134964-134975, 2020, doi: 10.1109/ACCESS.2020.3005823.

[15] Y. Zhou, Y. Yang, H. Liu, X. Liu and N. Savage, "Deep Learning Based Fusion Approach for Hate Speech Detection," in IEEE Access, vol. 8, pp. 128923128929,2020,doi:10.1109/ACCESS.2020.3009244.

[16] P. S. Br Ginting, B. Irawan and C. Setianingsih, "Hate Speech Detection on Twitter Using Multinomial Logistic Regression Classification Method," 2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), 2019, pp. 105-111, doi: 10.1109/IoTaIS47347.2019.8980379.

[17] Brown Alexander. 2017. What is so special about online (as compared to offline) hate speech? Ethnicities (2017).

[18] EEANews. Countering hate speech online, Last accessed: July 2017, http://eeagrants.org/News/2012/.

[19] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, Antalya, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.