CNN based Hate-o-Meter: A Hate Speech Detecting Tool

Ajinkya Chaudhari
Department of Computer Engineering
Vishwakarma Institute of Technology
Pune, India
ajinkyajc@gmail.com

Akshay Parseja
Department of Computer Engineering
Vishwakarma Institute of Technology
Pune, India
akshayparseja@gmail.com

Akshit Patyal
Department of Computer Engineering
Vishwakarma Institute of Technology
Pune, India
akshit.patyal18@vit.edu

Abstract— Hate Speech is a widespread problem that degrades a person or people based on their race, religion, gender or disability. This research work proposes a tool to raise awareness on the persistent hate speech in blogs, online-forums, and newspapers. The primary aim of this research work is to highlight the content that promotes violence or hatred against individuals or groups based on religion, gender, ethnicity or disability. A convolutional neural network architecture is used along with the natural language processing techniques. Using this algorithm, the tool identifies the percentage of hate and displays the bias of the statements. To host the proposed model, flask API and heroku platform is used. The proposed tool has the ability to detect hate speech with 80.15 percent accuracy and f1-score of 80.35 percent. The tool is made free and available for demo use to the public.

Keywords — hate speech, CNN, word2vec, text, natural language processing

I. INTRODUCTION

In general, hate speech refers to any statement that causes spread of, promotes or incites hatred, violence or discrimination against a person or community. Moreover hate speech and hate-motivated violence remains as a major concern across the globe. Victims rarely report the incidents to the authorities for the fear of retaliation or of not being taken seriously, or because they have no confidence in the justice system. This results in the lack of availability of legal organization, due to which there is no action taken against the debaucher. Our tool's main purpose is to recognize any form of text that expresses hate. Our research showed us that the sources of hate speeches are usually news articles, online forums, and blogs. Through our interface, the user will be able to identify how much of hate that a text corpus portrays. In many countries, hate speech subjects to a freedom of speech, for example, United States does not have hate speech laws, according to the First Amendment to the U.S Constitution, the guarantee to freedom of speech is compromised due to the hate speech law. On the contrary, many countries classify hate speech as an illegal activity. India prohibits hate speech by several sections of the Indian Penal Code, the Code of Criminal Procedure, and many more laws placed in order to avoid hate crimes. Section 95 of the Code of Criminal Procedure gives the government the right to declare certain publications "forfeited" if the "publication appears to the State Government to contain any matter the publication of which is punishable under Section 124A or Section 153A or Section 153B or Section 292 or Section 293 or Section 295A of the Indian Penal Code". The Strategy and Plan of Action on Hate Speech helps and guides the United Nations system to come up with methods to tackle hate speech at the national and global level. It supports UN Secretariat to help the United Nations Resident Coordinators in addressing and countering hate speech. Our tool is designed to help such organizations and governments to achieve the common objective of combatting hate speech. Hate-O-Meter will aid them to find people or associations who are the nucleus of publishing hate statements on their platform. Our tool is available at: https://hate-o-meter.herokuapp.com/.

II. LITERATURE REVIEW

As hate statements are under reported, it was difficult to decide our dataset. There were minimal resources of data related to hate speech, especially 'Formal Hate Statements'. This paper [3] had worked on data from Stormfront and used web scraping and data pre-processing techniques to form a dataset. They included Hate, No Hate, Relation and Skip as annotations. This dataset was very well defined and annotated. It is also found that the sentences in the dataset were more formal and professional as compared to any other data related to hate speech. To achieve maximum results out of our work, it has used the data set provided by the authors of the paper [3]. Here the authors are credited for building such a dataset available in our model. Few research papers pertaining to our work are analyzed before building our own model. Our initial focus was on categorizing a single sentence as hate or not hate. It is learnt from this paper that a not efficient. The previously implemented natural language processing approach are reviewed to classify text statements using deep learning, where it was evident that the methods used could be improvised. Thus, we implemented Convolutional Neural Networks with optimized parameters for the task. It is found that, the paper [5] is significant for the fact that the authors used a simple bad of words approach with the logistic regression and SVM architectures, then compared these models with deep learning approaches. They were able to achieve an F-score of 84.83 using GloVe embeddings. Based on the proposed approach on similar lines and used the word2vec model for our purpose. The paper [5] motivated us to use Convolutional Neural Network for our tool. We experimented with our CNN model to come up with the best parameters and achieve high accuracy. The paper [6] introduces a Twitter hate-speech text classification system by comparing logistic regression and CNN techniques. The classifier used in the authors' research assigns each tweet to one of four predefined categories: racism, sexism, both and non-hate statements. They were able to achieve an F-score of 77.75% which could be increased. This study motivated us to research about category-bias of hate and learn that it could classify the text input into religion, gender, ethnicity and disability based on what words are included in the text.

III. METHODOLOGY

A. Theory:

a) Data:

As the complications are faced while finding suitable hate speech data sets, here a combination of three datasets are used as per the needs of the proposed research and implementation. Majority of our data contained twitter tweets. Thus, we had to pre-process our data by a self-defined python script using the regular expressions package. We removed all symbols except punctuation, removed all blank spaces before and after a sentence, removed any 'twitter usernames', and deleted sentences that were not significant. We used Aitor Garcia's data [3] which had formal hate statements with zero twitter statements, Waseem's data [7] of tweets, and added some headlines and relevant sentences from online forums. The final data set comprises single sentences labelled either hate i.e. 1 or non-hate i.e. 0. Table 1 describes the division of hate and non-hate statements. The model was trained on 13029 samples and validated on 4343 samples. The dataset can be accessed from: https://github.com/ajinkya-ch/Hate-o-Meter/blob/master/finaldata.csv

TABLE I – DATASET DESCRIPTION

Class	Number of Statements
Hate	6491
Non- hate	13414

b) Spacy:

This library allowed us to implement natural language processing techniques. Primarily, we used it to tokenize our sentences in the data file into words (tokens), then to convert all words to lower case, remove all stop words (words that do not define the meaning of a sentence). By this process as seen in Figure 1, we achieved our Vocabulary List. Furthermore, we used the package for dividing large articles/ paragraphs into sentences and words for an error- free prediction process.

c) Newspaper Package:

Newspaper is a library in Python while helped us to extract article content from any URL inputted. The content extracted goes through natural language processing and gets tokenized like any other text. We have used this as it is a feature of our tool and provides user to input URL instead of regular text.

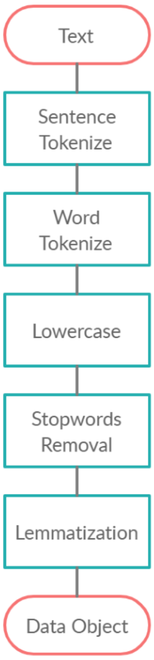


Fig. 1 Text Processing

d) Word2Vec Embedding:

We used the Google News Word2Vec model [8]. It provides pre-trained word embeddings for training our model. It turns words to vectors for our model to be trained on. It includes two algorithms, the bag-of-words and skip-gram approach. Both algorithms learn the representation of a word that is useful for prediction of other words in a sentence. The quality of word vectors depends on amount of data, quality of data, size of vectors, and main algorithm. We calculated embeddings according to our training vocabulary, which was largely based on news statements and headlines. As the model was built on a large dataset of news too, we could train our model with ease.

e) Keras API:

We have used the Keras functional API due to its ease to use, modular nature, and adaptation. It is built on the 'TensorFlow' backend [9]. Firstly, it is used to pre-process data by its Tokenizer. We were able to transform each word in the sentence to numbers/ vectors for the CNN model to understand. Each word is assigned an integer and that integer is placed in a list. For example, we have a sentence- "I like cats and dogs". Each word is assigned a numeric value; I=1, like=2, cats=3, and=4, dogs=5, and our MAX_SEQUENCE_LENGTH=10, then padding makes the sentence=[0,0,0,0,0,1,2,3,4,5]. We are using the functional model API provided by Keras to build our CNN model. This model helped us to create a more complex model, in which layers could be interconnected better.

f) Flask API:

We deployed our CNN model onto the Flask environment as an API, to get visual representation of the working of our algorithm. The API loads the CNN model as a. Hdf5 file and the tokenizer as a pickle object. To predict the input provided to our rendered template using one of the two ways of entering data (raw data or URL) to get the outcome. When the raw text radio button is selected the user is being asked to provide the API with a text paragraph. When URL is to be scrapped the user can select URL radio button and then proceed with providing URL for which user wants to find hate speech percentage. We used CanvasJS to display output in graphical format.

g) Heroku Platform:

Heroku is a PaaS (platform as a service) known for running apps over a cloud platform. It can be scaled up or down based on the size of the application. There are many cost-free application hosting sites like 'Python Anywhere', but Heroku is used as a better option for being quite robust and provides the average storage amount for a large model. Heroku has its own GIT repository that makes it even more simple for deploying the applications.

B. Algorithm:

We approached the problem of detecting and classifying hate speech with the 'Convolution Neural Network' method which is an artificial neural network based on shared-weights architecture and translation invariance characteristics. The structure of the model is defined below:

- Input type: The user inputs either text (sentence or paragraph) or an URL.

a) Single sentence: Gets stored in a data frame first. Then the data frame as a list gets passed into the Keras tokenizer for as it is.

b) Paragraph: First through natural language processing the paragraph is divided into multiple sentences using Spacy (Section III-A- 2). The sentences are stored in a data frame which gets passed into the tokenizer as a list.

c) URL: If a user inputs a website URL, the newspaper API (Section III-A-3) will first extract all content. Then, it goes through a process like (b).

After the Keras tokenizer gets sentence input, it converts them to vectors, then passes the sequence into the model.

- Convolutional Neural Network model:

a) Input Layer: Takes the hot encoded form of text corpora in any of the above input type

b) Embedding: The embedding weights are provided along with the maximum sequence length. Each token of the sentence is mapped to its corresponding word embedding, from the Word2Vec model.

c) Convolutional Layers: We used 200 filters of sizes 2,3,4,5,6. The layers use 'relu' activation. These layers help to apply units across the text sentences.

d) GlobalMaxPooling1D Layers: We concatenated 5 maxpooling layers with the convolutional layers. To this concatenated layer we applied a dropout of 0.1. These layers define the tensor shape for the sentence according to the max vector in the sentence.

e) Dense Layer: We used a dense layer with an activation of 'Leaky ReLU', alpha equal to 0.3 and a dropout of 0.2 to overcome the 'Dying ReLU problem'.

f) Output Layer: We used activation function = 'Sigmoid' as our problem statement was aimed to binary classify.

- Hate Category Calculation:

We stored words related to Religion, Gender, Ethnicity and Disability into respective lists. After applying natural language processing on the text, we checked if the token is any of list. If yes, then we display it through graphical representation.

- Output:

The model predicts and classifies the given sentence(s) into 1 or 0, based on hate or non- hate. We iterate over the predictions for every sentence. And using the formula mentioned in (V), calculate the percentage of hate.

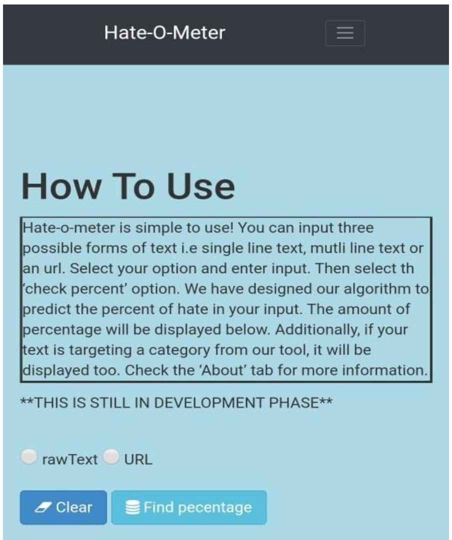
IV. RESULTS AND DISCUSSIONS

With this Convolutional Neural Network architecture, we were able to achieve an accuracy of 80.15 percent. We compiled the model using Adam Optimizer and after comprehensive analysis of our method, we achieved the results as stated In Table 2. When our model predicts a given text sentence as hate, it is accurate 79.5% of the time. The model can rightly predict 81.1% of the statements. Thus, our model is successful in identifying hate percent in text corpora and its bias. In [6] the authors used a convolutional model to classify hate speech and achieved a 78.3% F-score. In [10], context aware models were used but could only attain a maximum F-score of 60%. Thus, by using a optimized CNN approach, we achieved the metrics mentioned in Table II.

TABLE II- EVALUATION METRICS

Precision	0.7955
Recall	0.8119
F1- score	0.8035

The front page of the API is seen in Figure 3. The authors have designed the tool to implement the CNN model and provide an interactive method of testing the amount of hate in some text.



How To Use

Hate-o-meter is simple to use! You can input three possible forms of text i.e single line text, mutli line text or an url. Select your option and enter input. Then select th 'check percent' option. We have designed our algorithm to predict the percent of hate in your input. The amount of percentage will be displayed below. Additionally, if your text is targeting a category from our tool, it will be displayed too. Check the 'About' tab for more information.

THIS IS STILL IN DEVELOPMENT PHASE

Fig. 2. Front Page of Tool

In Figure 2, we see the front and main page of our API. We are given a short message about the how the tool works, then we can either select to raw Text or URL as discussed in III-B. After our selection, an input box pops up, enter your data and click on 'Find Percentage'. As a result, you will see Figure 4.

A. Input Case*:* Button Selected- RawText

Text data: *'My love for India never stops. But hindus hate muslims here.'*

B. Expected Result*:*

a) Sentence Tokenizing:

Sentence 1: My love for India never stops.

Sentence 2: But hindus hate muslims here.

b) CNN classification:

Sentence 1: 0 (non hate)

Sentence 2: 1 (hate)

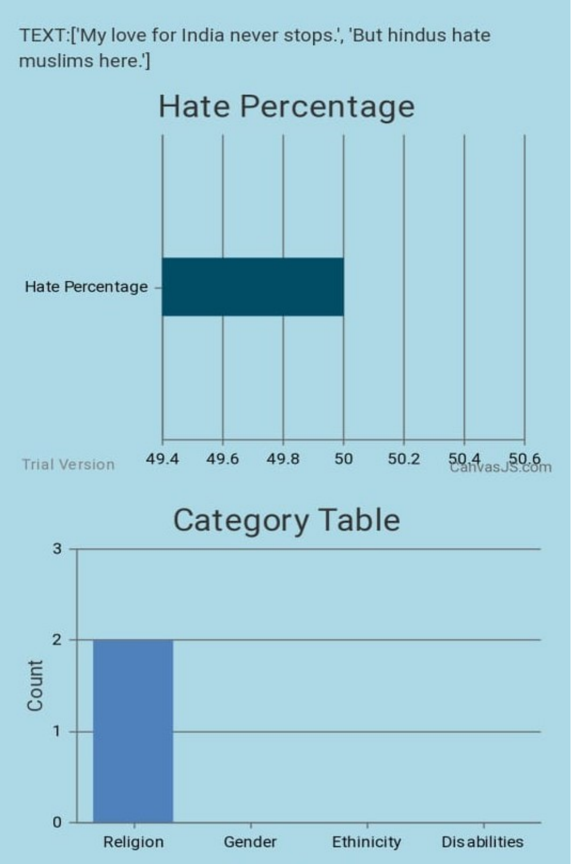
c) Hate Category Calculation:

Sentence 1: has no occurrences of any bias

Sentence 2: has two occurrences of the religion bias (hindus, muslims)

C. *Output:*

We see in Figure 4 that the hate percent calculated is 50% and displayed in graphical format. We can verify that by using formula in section (V). The Category Table displays the targeted hate- biases of the text input. In our case, Religion is the highest targeted with 2 related word occurrences in total.



TEXT:['My love for India never stops.', 'But hindus hate muslims here.']

Fig. 3. Output of Test Case

V. MATH

Hate Percent Formula:

$$Hate\ Percent = (Total\ hate\ statements\ in\ text \div Total\ statements\ in\ text) \times 100$$

VI. LIMITATIONS

The tool is modelled to detect formal hate speech, that in newspapers and articles. It is trained using formal statements and text. It is unable to detect sarcastic hate statements. The current model does not comprehend text comprised of a single sentence with less words, i.e. less than 6. The application cannot handle multiple requests in one instance. The user may be required to relaunch the API after a single request.

VII. FUTURE SCOPE

The authors aim to expand the project by adding 'multiple languages support' in the future, to help people who do not understand the English language. The method used to build the tool can be experimented and optimized for better results. The facility of crowdsourcing in our tool can be added, by which a person could report hate speech through the tool itself and the model will optimize itself. There is a room for improvement in the form of input to the tool. We can enable the tool to read text directly from .txt, .docx, .pdf files. As the tool can be used for legal data reporting purposes too, in the future, we can add a section which lists all the sources that are reported for their high hate speech content. This information will be valuable to legal organizations who investigate hate crimes. The authors welcome developers and researchers to further develop the tool. The code is available on: https://github.com/ajinkya-ch/Hate-o-Meter

VIII. CONCLUSION

The proposed work has developed a Hate-o-Meter to recognize any form of hate statements that are published through internet forums, newspapers, articles and blogs. The model is built using convolutional neural networks and natural language processing for the tool, that was able to achieve a commendable accuracy. There has not been any similar software developed in the past. The proposed model was able to achieve an F-score of 80.35%, which is better than the previously implemented methods. The tool is aimed to help any member for academic, personal, professional, or legal work. The authors hope that the tool may be used to reduce and accentuate any form of cyber-bullying, terrorism or hate in the future.

ACKNOWLEDGMENT

We are thankful to Professor A.S Shingare for guiding the authors during the development of the tool. We appreciate the efforts taken by her to guide us and help us to understand other studies in depth. We are grateful for our university, Vishwakarma Institute of Technology, which provided us with resources and knowledge that helped us to do our work. Finally, we are thankful for the endless support of our parents and peers.

REFERENCES

[1] Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate Speech Detection with Comment Embeddings. In Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion). Association for Computing Machinery, New York, NY, USA, 29–30.

[2] Mainack Mondal, Leandro Araújo Silva, and Fabrício Benevenuto. 2017. A Measurement Study of Hate Speech in Social Media. In Proceedings of the 28th ACM Conference on Hypertext and Social Media (HT '17). Association for Computing Machinery, New York, NY, USA, 85–94.

[3] Ona de Gibert, Naiara Perez, Aitor García-Pablos, Montse Cuadros. Hate Speech Dataset from a aWhite Supremacy Forum. arXiv:1809.04444[cs.CL]

[4] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Learning for Hate Speech Detection in Tweets. In Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 759–760.

[5] Paul, S. and Bhaskaran, J., ERASeD: Exposing Racism And Sexism using Deep Learning.

[6] Gambäck, Björn, and Utpal Kumar Sikdar. "Using convolutional neural networks to classify hate-speech." In Proceedings of the first workshop on abusive language online, pp. 85-90. 2017.

[7] Waseem, Zeerak, and Dirk Hovy. "Hateful symbols or hateful people? predictive features for hate speech detection on twitter." Proceedings of the NAACL student research workshop. 2016.

[8] Zhang Z., Robinson D., Tepper J. (2018) Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In: Gangemi A. et al. (eds) The Semantic Web. ESWC 2018. Lecture Notes in Computer Science, vol 10843. Springer, Cham

[9] N. A. Setyadi, M. Nasrun and C. Setianingsih, "Text Analysis For Hate Speech Detection Using Backpropagation Neural Network," 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, Indonesia, 2018, pp. 159-165.

[10] Gao, Lei and Ruihong Huang. "Detecting Online Hate Speech Using Context Aware Models." RANLP (2017).

[11] Zhang, Ziqi and Luo, Lei. 'Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter'. 1 Jan. 2019 : 925 – 945.

[12] Golbeck, Jennifer et al. "A Large Labeled Corpus for Online Harassment Research." WebSci '17 (2017).

[13] Pitsilis, Georgios K., Heri Ramampiaro, and Helge Langseth. "Effective hate-speech detection in Twitter data using recurrent neural networks." Applied Intelligence 48.12 (2018): 4730-4742.

[14] Wang, Cindy. "Interpreting neural network hate speech classifiers." Proceedings of the 2nd Workshop on Abusive Language Online (ALW2). 2018.

[15] Setyadi, Nabiila Adani, Muhammad Nasrun, and Casi Setianingsih. "Text Analysis For Hate Speech Detection Using Backpropagation Neural Network." 2018 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC). IEEE, 2018.

[16] Bashar, Abul. "Survey on evolving deep learning neural network architectures." Journal of Artificial Intelligence 1, no. 02 (2019): 73-82.

[17] Joby, P. P. "Expedient Information Retrieval System for Web Pages Using the Natural Language Modeling." Journal of Artificial Intelligence 2, no. 02 (2020): 100-110.