JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

NEGATIVE SPEECH DETECTION USING RECURRENT NEURAL NETWORK

[1] Katroth Balakrishna Maruthiram, [2]Vankudoth Vinod Kumar, [3]Bhookya Rajkumar

[1] Assistant Professor of CSE, JNTU, [2] M.Tech Student of JNTU, [3]Assistant Professor of CSE,
[1, 2] Department of IT, JNTU Hyderabad, India, [3] MLR Institute of Technology, Hyderabad, India

Abstract: The swift growth in internet usage has led to the emergence of multiple cyber issues, including negative speech and cyberbullying. Creating tools that can recognize foul language is an often used strategy to deal with the problem. We suggest applying Recurrent Neural Networks (RNNs) to the detection of hate speech. In the realm of natural language processing (NLP), RNNs—particularly Long Short-Term Memory (LSTM) networks—have proven to be highly efficient in handling word and paragraph sequences.

When generating subsequent predictions, RNN algorithms' memory capacities allow them to understand long-term dependencies in the data and the broader context of a sequence. Recurrent neural networks have been used in previous research for this purpose, and the results have shown that they can attain an accuracy of about 86%, which is higher than that of the Random Forest classifier (72%), and the AdaBoost classifier (70.8%). Recurrent Neural Networks are employed in our solution to identify hate speech and categorize material as either hateful or non-hateful.

IndexTerms **– Recurrent Neural Network (RNN), LSTM (Long Short Term Memory).**

I. INTRODUCTION

Global involvement has expanded in the last few decades due to social networks' broad usage. People now have the chance to widely express and share their opinions in real time thanks to microblogging services. Researchers can investigate online social emotions across a range of events with these platforms. Open communication between people has made it easier for a wide range of ideas, feelings, and knowledge to be shared. Online environments aren't always safe, though, and they can help spread offensive and hostile content. Hate speech is a pervasive online manifestation of prejudice and aggressiveness that includes verbal abuse, xenophobia, racism, and other forms of prejudice.

Speech that disparages people or groups because of their race, ethnicity, sexual orientation, gender, religion, or nationality is generally referred to as hate speech. Social media sites receive a massive amount of posts and comments every second, which makes it difficult to properly monitor or control content. As a result, these platforms find it difficult to restrict these kinds of posts while yet upholding free expression. Hate speech can spread due to the diversity of individuals and their backgrounds, cultures, and beliefs. Furthermore, every culture has its own set of norms and beliefs that influence how it views and deals with cyber-hatred.

Cyber-hate is a term used to describe the occurrence of hate speech and violent communication on the internet. It is the use of electronic communication tools to spread messages that are racial, religious, extreme, or terrorist in nature. It is a subcategory of cyberbullying. Dissimilar to cyberbullying, which generally targets single people, hate speech has the potential to impact entire communities. Any written or spoken expression that promotes violence against anyone on the basis of their race, ethnicity, nationality, religion, sexual orientation, or gender identity is considered hate speech, according to Brown. Hate speech can take many different linguistic forms and include derogatory remarks, threats, abuse, and violence.

II. PURPOSE

The increase in hate speech on social media in recent years has led to significant attempts by businesses, governments, and academics to identify workable remedies. Many methods have been developed to automatically identify hate speech on the internet. Rapid growth of internet forums and Twitter has revolutionized communication and content sharing, but it has also been used to spread hate speech and organize actions motivated by hatred. These platforms' accessibility and anonymity encourage the production and spread of hate speech, which can eventually result in hate crimes online and pose a challenge to established law enforcement strategies.

The first step in developing effective countermeasures for hate speech on the internet is to identify and monitor such information. Every year, social media behemoths like Facebook, YouTube, and Twitter have been contributing hundreds of millions of euros to this project. They continue to be criticized for their alleged insufficiency in spite of these large expenditures. This is mostly due to the fact that their efforts mostly depend on manual moderation, which is a labor-intensive, time-consuming, and ultimately unsustainable and unscalable method of identifying and eliminating offensive content.

III. EXISTING SYSTEM

Over the past few years, researchers have been examining hate speech on social media sites like YouTube, Reddit, and Twitter. The features employed are what primarily set traditional machine learning techniques apart; the most predictive features are surface-level features like bag of words, word-level, and character-level n-grams. In addition to features, many methods have been used for classification, including Support Vector Machines, Naive Bayes, and Logistic Regression. Davidson et al. used crowdsourcing to classify a 24K tweet corpus containing hate speech keywords as either hate speech, offensive language, or neither. A variety of features were retrieved, including Part of Speech (POS) labelling, n-grams, and tweet-level metadata (amount of hashtags, mentions, retweets, and URLs). Logistic Regression using L2 regularization turned out to be the most effective multi-class classifier in their testing.

An ensemble-based method that uses many linear SVM classifiers in tandem to separate hate speech from ordinary profanity on social media was proposed by Malmasi et al. Djuric et al. presented a two-step approach as one of the first attempts to use neural network models. They used a continuous bag of words model to extract paragraph2vec embeddings and a binary classifier trained in conjunction with the embeddings to distinguish between clean and hateful text. In this study, Badjatiya et al. investigated three deep learning architectures: CNN, LSTM, and FastText. They initialized the word embeddings using either GloVe or random embeddings. Based on a CNN model trained on a variety of feature embeddings, including word embeddings and character n-grams, Gambäck et al. created a hate speech classifier. Zhang and colleagues employed a CNN+GRU (Gated Recurrent Unit) neural network model that was trained using pre-trained word2vec embeddings to obtain word/character dependencies (order information) as well as word/character combinations (e.g., n-grams, phrases).

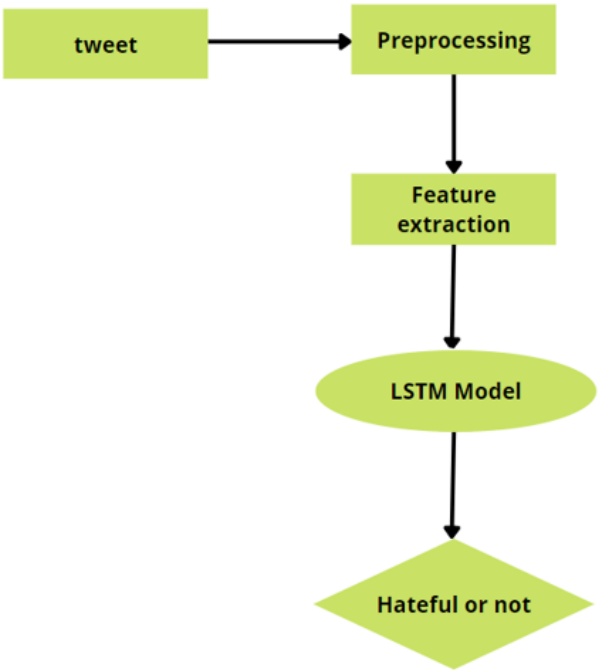
IV. PREPOSED SYSTEM

Negative speech on social media can lead to "cyber conflict," which has an impact on people's social lives both locally and nationally. Aside from having the capacity to seriously hurt people, hateful and antagonistic content distributed via social networks can also fuel social unrest and conflict outside of online. Social networks, however, are powerless to regulate every piece of user-posted content. As a result, the need for automatic hate speech identification is rising.

Negative speech is difficult to understand since it is subjective and subtle, and it varies greatly between cultures. These qualities are frequently disputed since what one person perceives as pure hatred may not be so to another. Social media and the internet are growing rapidly, and this has made people more reactive. In order to provide a secure environment for all users, the project's goal is to develop a model that can identify offensive information on social media sites, particularly Twitter.

V. PROJECT FLOW

This module introduces the steps involved in negative speech detection.



VI. BUILDING THE RECURRENT NEURAL NETWORK

6.1 Long Short-Term Memory

An improved variety of recurrent neural network (RNN) called the Long Short-Term Memory (LSTM) Network was created expressly to solve the problem of vanishing gradients that conventional RNNs had. lengthy-term dependencies in sequential data are easily captured and retained by LSTMs, in contrast to typical RNNs that suffer from the vanishing gradient problem when trying to retain information over lengthy sequences.

They accomplish this by adding specialized units that have the ability to selectively retain or forget information over time. This makes them ideal for jobs requiring persistent memory and efficiently managing long-term dependencies. Every recurrent neural network is composed of a series of neural network modules that repeat. This repeating module in conventional RNNs will have a very basic structure, like a single tanh layer.

This chain-like structure is also present in LSTMs, albeit the repeating module is structured differently. There are four neural network layers instead of just one, and they interact in a unique way. Every line in the diagram below represents a vector that flows from one node's output to other nodes' inputs. The learnt neural network layers are represented by yellow boxes, and pointwise operations, like vector addition, are represented by pink circles. Lines that merge together signify the concatenation of vectors, whereas lines that fork suggest the copying of a vector's content to multiple locations.

6.2 The Core Idea Behind LSTMs

The horizontal line across the top of the diagram represents the cell state, which is an important feature of LSTMs. It spans the whole chain with little linear interaction, much like a conveyor belt. The ability of the cell state to retain and transmit information over extended periods is due to its ability to transfer information without interruption. With the help of structures known as gates, the LSTM is able to carefully control what information is added or removed from the cell state.

Information can be optionally allowed via gates. They consist of a pointwise multiplication operation and a sigmoid neural net layer .Numbers ranging from zero to one are produced by the sigmoid layer, which indicate how much of each component should pass through. A number of one indicates that "everything through," whereas a value of zero indicates "let nothing through." Three of these gates are present in an LSTM to safeguard and regulate the cell state.

6.3 Step by Step LSTM Walk Through

Determining what data to discard from the cell state is the first stage in our LSTM. The "forget gate layer" is a sigmoid layer that makes this determination. After examining ht−1 and xt, it generates a number in the range of 0 to 1 for every number in

the cell state Ct−1. "Completely keep this" is represented by a 1 and "completely get rid of this" by a 0.Consider a language model that attempts to forecast a word's future based on every word that has come before it. To ensure that the appropriate pronouns are used in this situation, the gender of the current subject may be included in the cell state. We want to forget the gender of the previous subject when we view a new one.

Choosing the new data to be stored in the cell state is the next stage. There are two components to this. First, the values to be updated are selected via a sigmoid layer known as the "input gate layer." A vector of potential new values, C^t, that could be added to the state is then produced by a tanh layer. We'll combine these two to produce a state update in the following step.
To replace the old subject we're forgetting, we would want to add the gender of the new subject to the cell state in our language model example.The previous cell state, Ct−1, needs to be updated into the new cell state, Ct. Everything has already been decided in the preceding steps; all that remains is to carry it out.

We multiply the previous state by foot, disregarding the information we previously chose to ignore. Next, we include it∗C~t. This represents the updated candidate values, adjusted for the amount that we chose to update each state value. As we decided in the previous steps, in the case of the language model, this is where we would actually add the new information and remove the information about the gender of the old topic.Ultimately, we must choose what we will produce. This will be a filtered version of the output, based on our cell status. The sigmoid layer is executed first, and it determines which portions of the cell state we will output. After that, we multiply the cell state by the sigmoid gate's output after running it through tanh to force the values to be between -1 and 1. This allows us to output only the portions that we have chosen.

Given that the language model example has just encountered a subject, it may wish to output data pertaining to a verb in the unlikely event that such is what will happen next. It may output, for instance, whether the subject is singular or plural, allowing us to determine the appropriate conjugation form for a verb in case that is what comes next.

IV. RESULTS

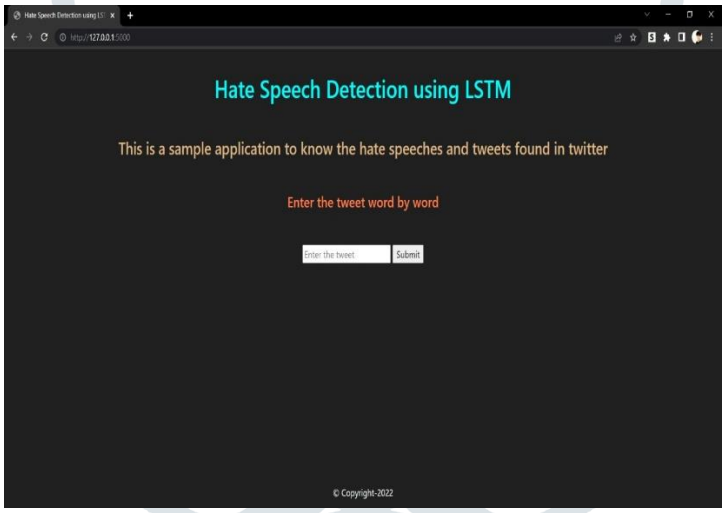


Figure showing frontend of the project

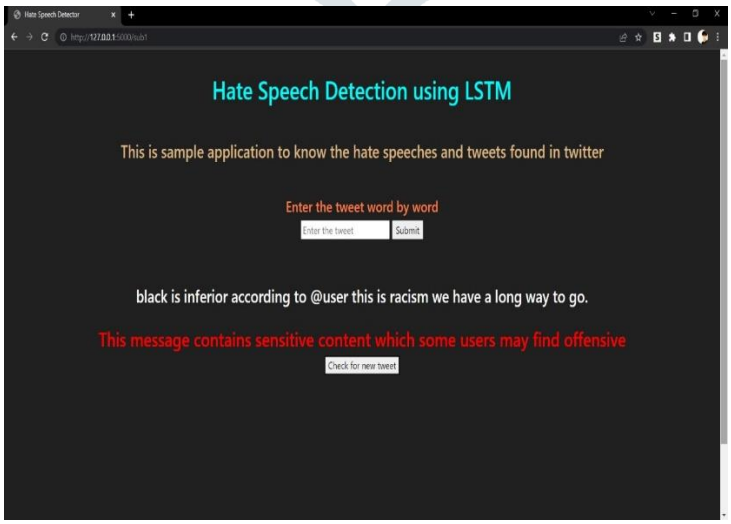


Figure showing the detection of negative speech yes,

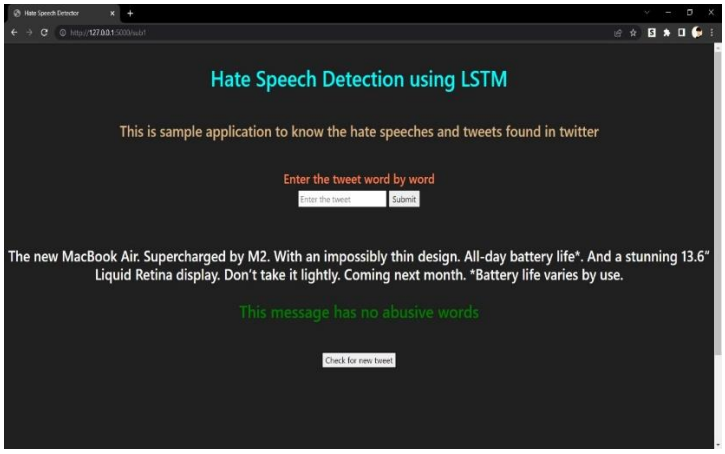


Figure showing the detection of negative speech no

VII. Conclusion

A classification report is used to assess how well a classification system predicted the world. the percentage of forecasts that are right or wrong. More specifically, True Positives, False Positives, True Negatives, and False Negatives are used to forecast the metrics of a classification report.The goal of this project was to address the growing concern of online toxicity by constructing a recurrent neural network (RNN) based hate speech identification system. The main goal was to accurately classify hate speech by utilizing RNNs' capacity to detect sequential dependencies in text data. After thorough testing and analysis, our RNN-based model showed encouraging results in differentiating between hate speech and non-hate speech. The model successfully picked up on linguistic patterns and contextual subtleties, which are necessary for reliable identification in a variety of online content.

Important conclusions from our research highlighted how well RNN architectures—like the Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM)—handle the sequential structure of text data. By lowering noise and improving data quality, preprocessing methods like text normalization and feature engineering greatly increased model performance.

Notwithstanding these developments, problems with hate speech detection—such as disparities in class and context—remain. Resolving these issues could improve automated moderation systems' scalability and dependability even more. To sum up, recurrent neural network use in hate speech detection is an important step toward creating safer online spaces. To reduce biases and improve the inclusivity of detection algorithms, future research topics should examine sophisticated deep learning techniques, multilingual approaches, and ethical considerations.This study emphasizes how important machine learning is in solving societal problems and how responsible AI solutions are necessary for online content monitoring.

References

[1] Analysis_Text_of_Hate_Speech_Detection_Using_Recurrent_Neural_Network

[2] Facebook, Google and Twitter agree german hate speech deal.

[3] chmidt, R.M.: Recurrent neural networks (RNNs): a gentle introduction and overview. arXiv preprint arXiv:1912.05911

[4] Gandhi I, Pandey M (2015) Hybrid ensemble of classifiers using voting. In: 2015 international conference on green computing and Internet of Things (ICGCIoT), pp 399–404. https://doi.org/10.1109/ICGCIoT.2015.7380496

[5] Joulin A, Grave E, Bojanowski P, Mikolov T (2016) Bag of tricks for efficient text classification. arXiv:http://arXiv.org/abs/1607.01759