Hate Text Detection in Chat Application

Yoga Anand K R , Thaitheya Sudan P K, Santhosh L B , Sasirekha C

Student, Student, Student, Assistant Professor
Dept. Of Electronics and Communication Engineering
K.L.N College of Engineering, Sivagangai, Tamilnadu, India

Abstract:
Online hate text is an important issue that breaks the cohesiveness of online social communities and even raises public safety concerns in our societies. There has been a lot of chatting applications present in recent years to connect with each other's through online like WhatsApp, Facebook, telegram etc. This paper proposes the techniques of creating a chat application that don't allow the user to send inappropriate messages. Before sending messages to the users, the typed message is evaluated whether it contains abusive words or not. If any, the API used here will detect that and censor the particular word and then warn the user. If the user uses cuss words beyond a certain limit, then he/she will be blocked permanently.

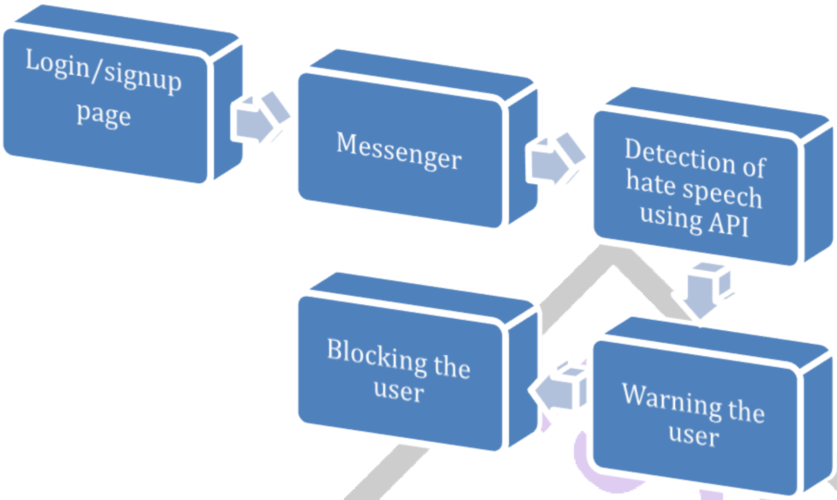
Keywords: **Hate text detection, cyber bullying**

Introduction:
Abusive language refers to any type of insult, vulgarity, or profanity that debases the target; it also can be anything that causes aggravation. Recently, an increasing number of users have been subjected to harassment, or have witnessed offensive behaviors online. Hate text detection is a process of detecting whether a sentence has hate in it or not. To provide a safe and supportive space for users is one of the key aspects of maintaining good and decent user experience on any platform. This means detecting and removing any comments or messages that may discriminate against or otherwise degrade a person based on their demographic information, as is the case with hate text. In this paper, we are going to build a custom full stack application that utilizes online hate speech detector. The application we will be building is a real-time chat application that is able to detect the hate text of the users' messages.To ensure that this process is completed in a timely manner, the following API will run an analysis on the text input to automatically detect if it contains hate text. With that, you can confidently provide a protected space for the users in any platforms.

Related Work:
The research community introduced various approaches on abusive language detection. Razavi et al. (2010) applied Naïve Bayes, and Warner and Hirschberg (2012) used Support Vector Machine (SVM), both with word-level features to classify offensive language. Xiang et al. (2012) generated topic distributions with Latent Dirichlet Allocation (Blei et al., 2003), also using word-level features in order to classify offensive tweets. More recently, distributed word representations and neural network models have been widely applied for abusive language detection. Djuric et al. (2015) used the Continuous Bag Of Words model with paragraph2vec algorithm (Le and Manolov, 2014) to more accurately detect hate speech than that of the plain Bag Of Words models. Badjatiya et al. (2017) implemented Gradient Boosted Decision Trees classifiers using word representations trained by deep learning models. Other researchers have investigated character level representations and their effectiveness compared to word-level representations (Mehdad and Tetreault, 2016; Park and Fung, 2017). As traditional machine learning methods have relied on feature engineering, (i.e. n-grams, POS tags, user information) (Schmidt and Wieg and, 2017), researchers have proposed neural-based models with the advent of larger datasets. Convolutional Neural Networks and Recurrent Neural Networks have been applied to detect abusive language,and they have outperformed traditional machine learning classifiers such as Logistic Regression and SVM (Park and Fung, 2017; Badjatiya et al., 2017). However, there are no studies investigating the efficiency of neural models with largescale datasets over 100K.

Methodology:

The high level application architecture consists of utilizing React and TypeScript for building out our custom user interface Using Node.JS and the Socket.IO library to enable real-time, bidirectional network communication between the end user and the application server. Since Socket.IO allows us to have event-based communication, we can make network calls to our ML services asynchronously upon a message that is being sent from an end user host.



The web app itself is quite simple and mostly just orchestrates everything. We decided to go with Node js for back-end, react js for front end and used mongo db as database. The implementation consisted of 1.Taking string as input. 2. Detect and censoring the hate-text using Machine learning API.

Node.js:

The chat app is mainly developed using JavaScript. Since JavaScript can only be able to run in client side and cannot be accessed through the server side. We cannot run JavaScript directly on a computer but JavaScript can run on a browser. This is because browsers contain a engine called “V8” engine that is written in C++ by google, that compiles JavaScript into a machine code. So by passing the JavaScript code through the V8 engine, the computer can understand JavaScript within the context of the browser. However, JavaScript cannot run outside the browser as there is no V8 engine outside the browser, so there node comes. Node js is a program that is written in C++. The V8 engine which is in the browser also lives inside node as well. By installing node it can read JavaScript and run it through the V8 compiler and convert it into the machine code so that the computer can understand. So we can now run JavaScript directly on computer and server and not just on browser. Node.js is an open-source environment and it’s free. It can run on various operating systems like windows, Linux, Mac OS etc. Node.js allows handling thousands of connection with a same server at a time without the burden of threading. It has a ideal advantage because not only helps the developer to develop front end but also now helps to handle the backend and server so it reduces the need of a developer to learn a different programming language for backend.

PHP vs Node.js in handling file request:

PHP: It first send a request to the computer and waits until the computer is back online and respond, then returns the result to the client and then only It will get ready for next request.

Node.js: Similar to PHP it also first send a request to the computer and then it will immediately get ready to handle the next request. When the server is online it returns the result to the client.

React js:

React is a open source JavaScript library. It is used for developing user interface in web pages. It is maintained by Meta which was formerly Facebook. React is used to develop single page websites, mobile applications.

React code is made of lot of components. Each component is responsible for reusable HTML code. Components play vital role in React applications. These components are formed in SRC folder naming in Pascal Case. Those components can be rendered in DOM using React DOM library. React can be installed easily using the npm command after installing NodeJS and NPM (Node Package Manager).

React library itself provides lot of additional features like emoji picker which is used to add emojis in the developing application, react toasts which is used to send alert/waring/danger messages to the users and so on.

Mongo db (data base):

MongoDB is the popular and mostly used NoSQL database; it is free and open-source document-based database. Mongo db is a document. It is composed of key and value pairs that are similar to JSON objects. Their values may be string or array. Mongo db stores the data in collections. The collection name will differ for each model. The storage format of mongo DB is called as BSON which is similar to JSON.

Features of MongoDB:

Document Based: MongoDB stores all the data in a same document instead of splitting into different documents.Indexing: Indexing is very important in a document that consists of database. MongoDB uses it to process huge data in small time.Availability: It increases the availability with many copies of data on different servers. When one server has got a problem and its down, the data can be recovered from other active servers which also have same data stored in that.Aggregation: In MongoDB computing also can be done and return the computed results. The few expressions are avg, sum, max, min etc.
Language support: MongoDB gives support for almost all the popular programming languages like C, C++, Node.js, PHP, Java, Python etc.

Express:

Express is a NodeJS framework. It can be imagined as a layer built on top of the NodeJS that helps to access or manage server and routes. It helps to build mobile or web applications easily. It can be used not only creating single page web application but also multipage and hybrid web applications. It helps to render HTML pages that are built using the extension named ejs by passing arguments to that ejs template.

Socket IO:

Socket.IO is a library that enables bidirectional communication between client and server:

1. **Features of socket IO:**

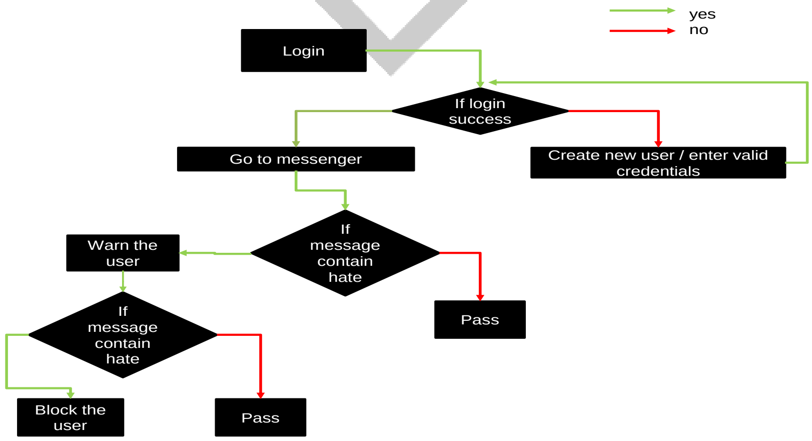
HTTP long-polling fallback: When the connection cannot be successfully Connected, the connection will fall back to HTTP long poling.

Automatic reconnection: Sometimes the WebSocket connection between client and server may be disconnected and the client may not know that disconnection occurred. In order to handle this Socket IO has a breathing mechanism which periodically checks the status of the connection of the server and client.

Packet buffering: While the Socket is not connected and a event is sent, the sent event will be buffering until reconnection. Once reconnected it will result in huge spike of events.

Broadcasting: On the server-side, one can send an event to all connected clients or to a subset of clients.

Flow Chart:



API used for this project:

Currently works with a mostly English database, the filter uses natural language processing (NLP) to decode the content into logical words ignoring punctuation, case, formatting, etc. We also apply word transformations to detect bad words by repeating characters, whitespace and special characters. Through detection and extraction of bad words you can also use this API to censor bad words from the detected text.

Security in our application:

In order to secure the user information from being hacked, hashing and salting are used. Hashing is a process of converting a key into a fixed length code that cannot be reversed to know the actual password and cannot be easily figured out by anyone.

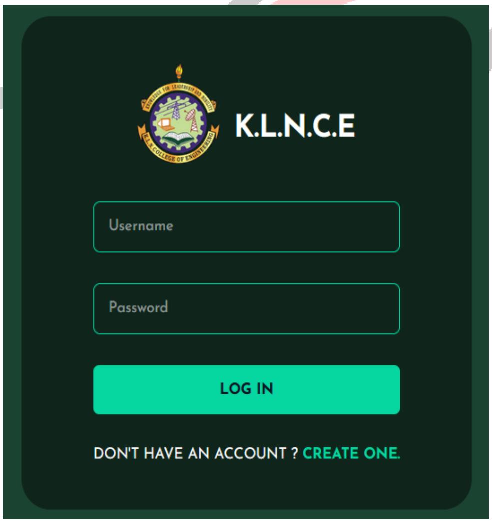
In addition to hashing, salting is used. A salt is a random string. By hashing a plain text password plus a salt, the hash algorithm's output is no longer predictable. The same password will no longer yield the same hash. The salt gets automatically included with the hash, so you do not need to store it in a database. Salting is the action of adding some random characters at the starting of user password and then converting it to hashed code. Salting provides an additional protection layer from getting hacked.

Working:

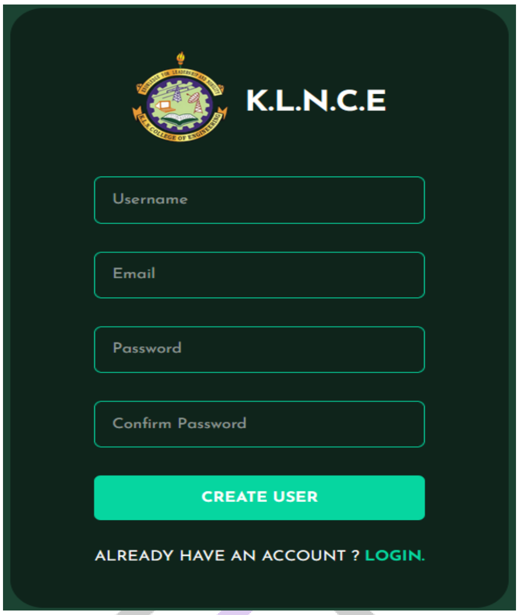
The working of this chat app is quite simple and more effective in live time chatting. The working starts with the account creation of the new user. Here, once the user opened the register page, the route which is handling the register function becomes active. Once the user entered all the details and submitted, it is verified with the data base whether the username/email is already present or not. If the username/email already exists it sends a toast alert notification which is provided by react else the user will be redirected to the chat page. Similarly the login page works. If the username or password is incorrect or did not exist a toast notification is sent to the user. The contact section consists of all the registered users. The chat page consist of a logout button on the top left corner, a input section on the bottom and a send button to send the messages. The input section consists of emoji picker that contains all the emojis. The send button and the emoji picker are provided by the react library and react makes everything simple. Once the message is sent, here comes our hate text detecting api into action. Every time before a message is sent, the message is checked using the api, if the message contains hate then the user will be warned for not being polite and the count of hate words usage will be modified in the database of the particular user who used hate words. If the user is getting toxic beyond a certain limit the user will be redirected to a warning page and will be banned permanently.

Model:

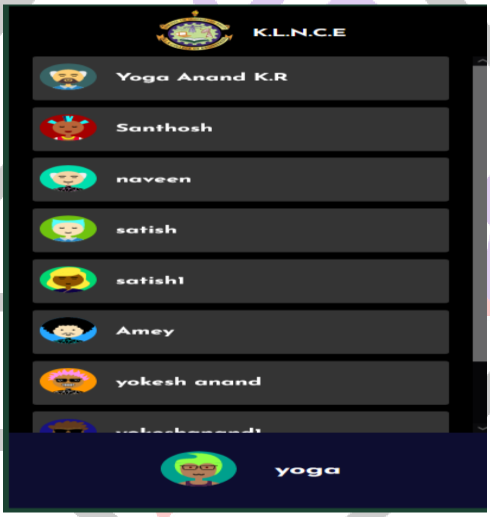
1 Login page:



2 Register Page:



3 Contact Page:



Yoga Anand K.R

Santhosh

naveen

satish

satish1

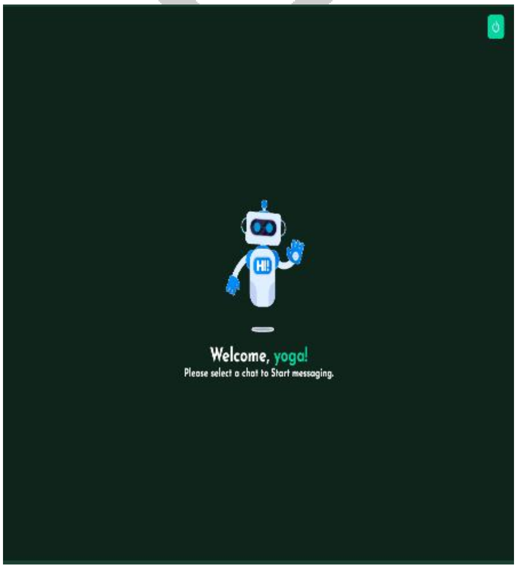
Amey

yokesh anand

yokeshanand1

yoga

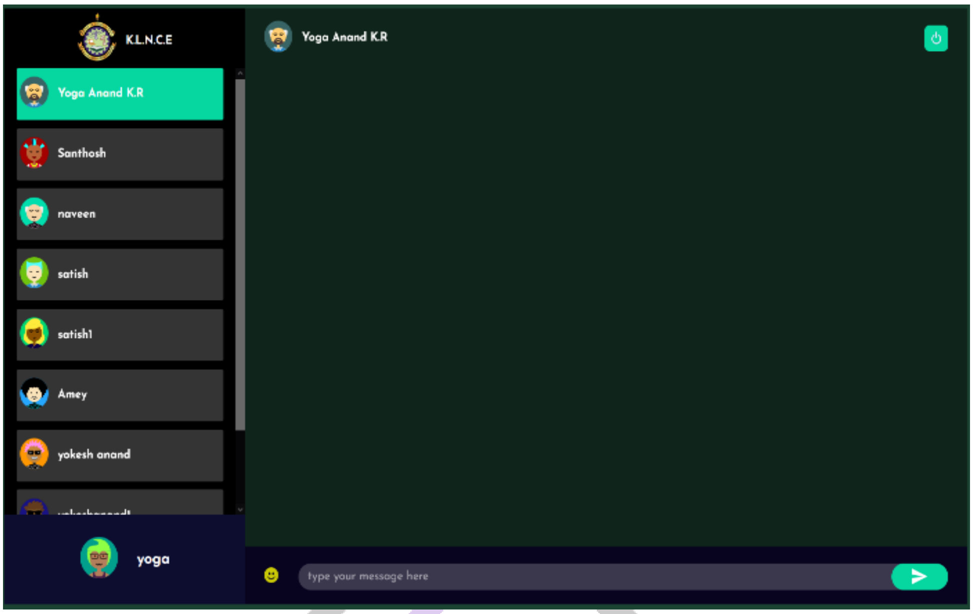
4 Welcome Page :



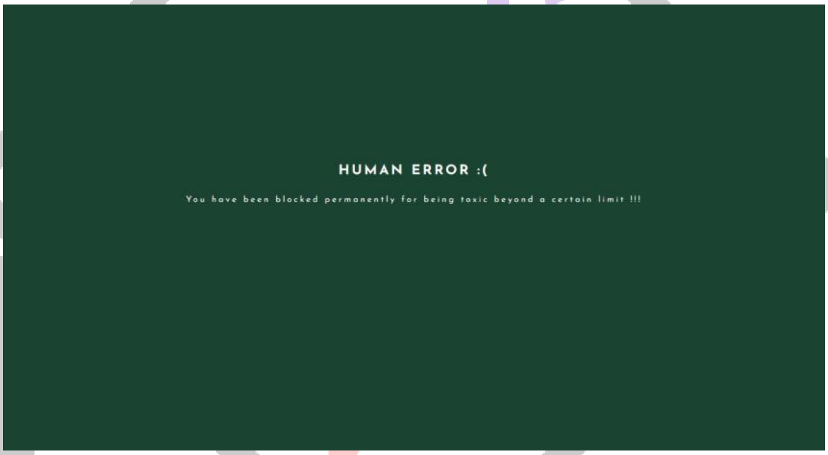
Welcome, yoga!

Please select a chat to Start messaging.

5 Chat Page:



6. Blocked page


HUMAN ERROR :(

You have been blocked permanently for being toxic beyond a certain limit !!!

Conclusion:

In this paper, we investigated the problem of hate text in various chat applications. We proposed a new chat application with features of blocking the users who uses abusive words. Internet has become the tool to amplify the hate speech phenomenon. To fight it, we must promote on the Internet the same rules and values that are the pillars of our society: diversity, tolerance and the respect of human rights.

References:

1. Amir H Razavi, Diana Inkpen, Sasha Uritsky, and Stan Matwin. 2010. Offensive language detection using multi-level classification. In Canadian Conference on Artificial Intelligence, Springer.
2. William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In Proceedings of the Second Workshop on Language in Social Media. Association for Computational Linguistics.
3. Castelle M. The linguistic ideologies of deep abusive language classification. In: Proceedings of the 2nd workshop on abusive language online (ALW2), Brussels; 2018.