CYBERBULLYING DETECTION SYSTEM IN SOCIAL MEDIA

Abstract:

Increasing the use of Internet and facilitating access to online communities such as social media have led to the emergence of cybercrime. Cyberbullying is very common now a days. which have no tracking like it may harm any individual, business, society, country in past few days it seems that riots were happened due to some statement used by one community on another its important to identify such content which spreads hate or harm community text processing, and machine learning algorithms such as Hybrid Cnn and Lstm .we are going to identify cyberbullying in twitter. Objectives of this implementation written in objective section. Image character with the help of OCR will be done by us to find image - based cyberbullying the impact on individual basis thus will be checked on dummy system. Machine learning and natural language processing techniques to identify the characteristics of a cyberbullying exchange and automatically detect cyberbullying by matching textual data to the identified traits. Supervised learning-based approaches typically use classifiers such as Hybrid CNN and Lstm to develop predictive models for cyberbullying detection.

Introduction

It is not sufficient to remind students of regulations forbid- ding plagiarism; In recent years, the use of social networking increased. And social networking sites are great tools of connecting to people. However, as social networking has become widespread. People are finding illegal and unethical ways to use these communities. We see that people, especially teens and young adults, are finding new ways to bully one another over the Internet. Bullying is not a new phenomenon and cyber bullying has manifested itself as soon as digital technologies have become primary communication tools. On the positive side, social media like blogs, social networking sites (e.g. Facebook), and instant messaging platforms (e.g. WhatsApp) make it possible to communicate with anyone and at any time. Moreover, they are a place where people engage in social interaction, offering the possibility to establish new relationships and maintain existing friendships. On the negative side however, social media increase the risk of children being confronted with threatening situations including grooming or sexually transgressive behaviour, signals of depression and suicidal thoughts, and cyberbullying. Users are reachable 24/7 and are often able to remain anonymous if desired: this makes social media a convenient way for bullies to target their victims outside the school yard. The detection of cyberbullying and online harassment is often formulated as a classification problem. Techniques typically used for document classification, topic detection, and sentiment analysis can be used to detect electronic bullying using characteristics of messages, senders, and the recipients. It should, however, be noted that cyberbullying detection is intrinsically more difficult than just detecting abusive content. Additional context may be required to prove that an individual abusive message is part of a sequence of online harassment directed at a user for such a message to be labelled as cyberbullying. The growth of cyberbullying activities is increasing as equally as the growth of social networks. Cyberbullying activities poses a significant threat to mental and physical health of the victims. Project about detection of bullying is present but implementation for monitoring social network to detect cyberbullying activities is less. Hence, the proposed system focuses on detecting the presence of cyberbullying activity in social networks using natural language processing and machine learning algorithms which helps government to take action before many users becoming a victim of cyberbullying. Detection of cyberbullying and the provision of subsequent preventive measures are the main courses of action to combat cyberbullying. The proposed method is an effective method to detect cyberbullying activities on social media. The detection method can identify the presence of cyberbullying terms and classify cyberbullying activities in social network such as Flaming, Harassment, Racism and Terrorism using natural language processing and machine learning algorithms. Cyberbullying detection is inherently difficult due to the subjective nature of bullying. It extends beyond detecting negative sentiments or abusive content in a message as these tasks, on their own, do not necessarily mean that the message is in fact bullying. For example, a message such as “I’m disgusted by what you said today and I never want to see you again” is difficult to classify as bullying without understanding the larger context of the exchange, even though the message is clearly expressing very negative sentiments. Conversely, positively-expressed sarcasm.

EXISTING SYSTEM:

Twitter is listed as one of the top five social media platforms where the maximum percentage of users experience cyberbullying (turbofuture.com, 2019). It enables a user to send a message of 280-characters, with more than 330 million active users at present (Statista, 2018). Studies on cyberbullying and Twitter often reported extensive cases of the phenomenon, with the potential for serious, deleterious consequences for its victims (Chatzakou et al., 2017a; Balakrishnan et al., 2019; Sterner, 2017). Several measures have been taken by Twitter to mitigate cyberbullying, such as filtering unwanted messages from users without a profile picture, and enabling a time-out feature that bans users using abusive language, among others. Despite these positive attempts, the platform is not completely immune from cyberbullying (Bernazzani, 2017; Twitter, 2019). Sentiments are the thoughts or opinions provoked due to the feelings attached with something, often categorized as positive, neutral or negative (Zhao et al., 2016). The process in which the unstructured data are computationally processed is referred to as sentiment analysis, and can be categorized into machine learning approach, lexicon-based approach and hybrid approach (Medhat et al., 2014). The machine learning approach employs algorithms such as Hybrid cnn and Lstm.

DIS ADVANTAGES :

1. Accuracy will be low. The time complexity is very high because we are working on text data.

2. Covert the text data into numeric form is very big task. We have to text preprocessing like removing the stop words, punctuation marks...etc.

3. Time consuming and prediction is not perfect

METHODOLOGY:

The proposed application should be able to detect the bullyed tweets. The feature representation is done by logistic regression, to convert To implement the model the steps to be followed are5

Dataset preparation and preprocessing

- Featuraization
- Data splitting
- Modeling Evaluation
- Hyper parameter Tuning
- Model Testing

PROPOSED SYSTEM:

In this paper, a solution is proposed to detect twitter cyberbullying. The main difference with previous research is that we not only developed a machine learning model to detect cyberbullying content but also implemented it on particular locations real-time tweets using Twitter API. The entire approach to detect and prevent Twitter cyberbullying is divided into 2 major stages: developing the model and experimental setup.

ADVANTAGES:

1. Cyberbullying research has often focused on detecting cyberbullying „attacks‟ and hence overlook other or more implicit forms of cyberbullying and posts written by victims and bystanders. However, these posts could just as well indicate that cyberbullying is going on

2. Speed and very low complexity, which makes it very well suited to operate on real scenarios.

OBJECTIVE:

Ability to detect cyberbullying through the use of online platforms is becoming increasingly important. Because there is too much information that humans cannot track, automatic detection that can identify threatening situations and hazardous content is required. This enables large-scale social media monitoring.

System Specification

Hardware Specification

Minimum system hardware requirement will be sufficient to execute the application since it does not take large or complex processing.

1) Hard Disk : 20 GB

2) RAM : 2 GB RAM

3) Processor Speed : 1.2 GHz

4) Processor : Pentium IV Processor

Software Specification

The other required software needed for development and testing of the software’s are

Operating System : Windows 2000 / XP, Linux based systems

Languages / Software: Python

Anaconda

Jupyter Notebook

MongoDB

Table: Literature Survey

Title	Authors	Problem	Solution	Result
An Effective Approach for Cyberbullying Detection and avoidance	Divyashree, Vinutha H, Deepashree N S	The biggest problem regarding cyberbullying is that the age group of the offenders ranges from as young as eight to the legal adult age of eighteen and beyond. Once happen this activity then victims are	In this paper focused on the issues of robust system and objectives are 1) Automatic detection and avoidance of cyberbully attack in internet. 2) Effective age authentication for	represented a novel method on the current scenario of cyber-bullying and various methods available for the detection and prevention of cyber harassment. Our concept depends upon the

		often left permanently then difficult to find them.	website browsing and categorizing the links based on age. 3) Effective website filtering in search results based on ranking. 4) Enhanced searching procedure promisingly reduces the effort of user in searching indented websites.	text analysis, the data which is uploaded or text written by any user is first analyzed.
Using Machine Learning to Detect Cyberbullying	Kelly Reynolds, April Kontostathis, Lynne Edwards	teens and young adults, are finding new ways to bully one another over	Used machine learning algorithm to detect	used a language-based method of detecting cyberbullying. By

		the Internet. in a study conducted by Symantec reported that, to their knowledge, their child has been involved in a cyberbullying incident.	cyberbullying. For training the data downloaded from website. The data was labeled using a web service. the labeled data, in conjunction with machine learning techniques provided by the Weka tool kit, to train a computer to recognize bullying content.	recording the percentage of curse and insult words within a post.
Cyberbullying Detection System on Twitter	Liew Choong Hon, Kasturi Dewi Varathan	Increased cyberbullying attacks on the social network services. To	this system, the users can identify the cyberbullying related tweets based on the	with the advent of this cyberbullying detection and solution system in Twitter, it will

		prevent these activities proposed an system.	keywords and populate it in a news feed form. By doing this, it allows users to determine the identities of the cyberbullies and the victims from the cyberbullying tweets	help the authorities to monitor, regulate or at least decrease the harassing incidents in cyberspace
Automatic detection of cyberbullying in social media text	Cynthia Van Hee,Gilles Jacobs,Chris Emmery,Bart Desmet,	Increased the cyberbullying using Social media sides/apps.	The focus of this paper is on automatic cyberbullying detection in social media text by modelling posts written by	In this paper investigate the automatic detection of cyberbullying-related posts on social media. Given the

			bullies, victims, and bystanders of online bullying. In this paper support vector machine is used to exploiting a rich feature set and investigate which information sources contribute the most for the task.	information overload on the web, manual monitoring for cyberbullying has become unfeasible. Automatic detection of signals of cyberbullying would enhance moderation and allow to respond quickly when necessary.
Methods for detection of cyberbullying: A survey	Rekha Sugandhi, Anurag Pande, Siddhant	major problem when it comes to cyber bullying is the lack of	This paper aims to review the different methods and algorithms	In this paper realize support vector machines have given the

	Chawla, Abhishek Agrawal, Husen Bhagat	identifiable parameters which mark any post as a bullying instance.	used for detection in cyber bullying and provide a comparative study amongst them so as to decide which method is the most effective approach and provides the best accuracy.	best result. We plan to implement SVM in our project as the primary classifier for our base dataset.

Aim of the Project

The main aim of the detecting the cyber bullying model will help to improve manual monitoring for cyber bullying on social networks. In this project we fetch the tweets from twitter accounts and preprocess the twits and images and applying generated model will detect the cyberbullying or not.

The objectives of the systems development and event management are:

Collect the dataset of bullying words and preprocess it and apply natural language processing and then machine learning algorithms Generate different machine learning algorithm model.

Fetch the tweets from twitter account and preprocess it.

Apply generated model on the fetched tweets and get final output cyberbullying or not.

Scope of the Project

Cyberbullying is the use of electronic communication to bully a person by sending harmful messages using social media, instant messaging or through digital messages. Cyberbullying can be very damaging to adolescents and teens. It can lead to anxiety, depression, and even suicide. Also, once things are circulated on the Internet, they may never disappear, resurfacing at later times to renew the pain of cyberbullying. Cyberbullying can be very damaging to adolescents and teens. It can lead to anxiety, depression, and even suicide. Also, once things are circulated on the Internet, they may never disappear, resurfacing at later times to renew the pain of cyberbullying. So overcome these issues detecting the cyberbullying is very important in now a days which will help to stop cyberbullying on social media networks.

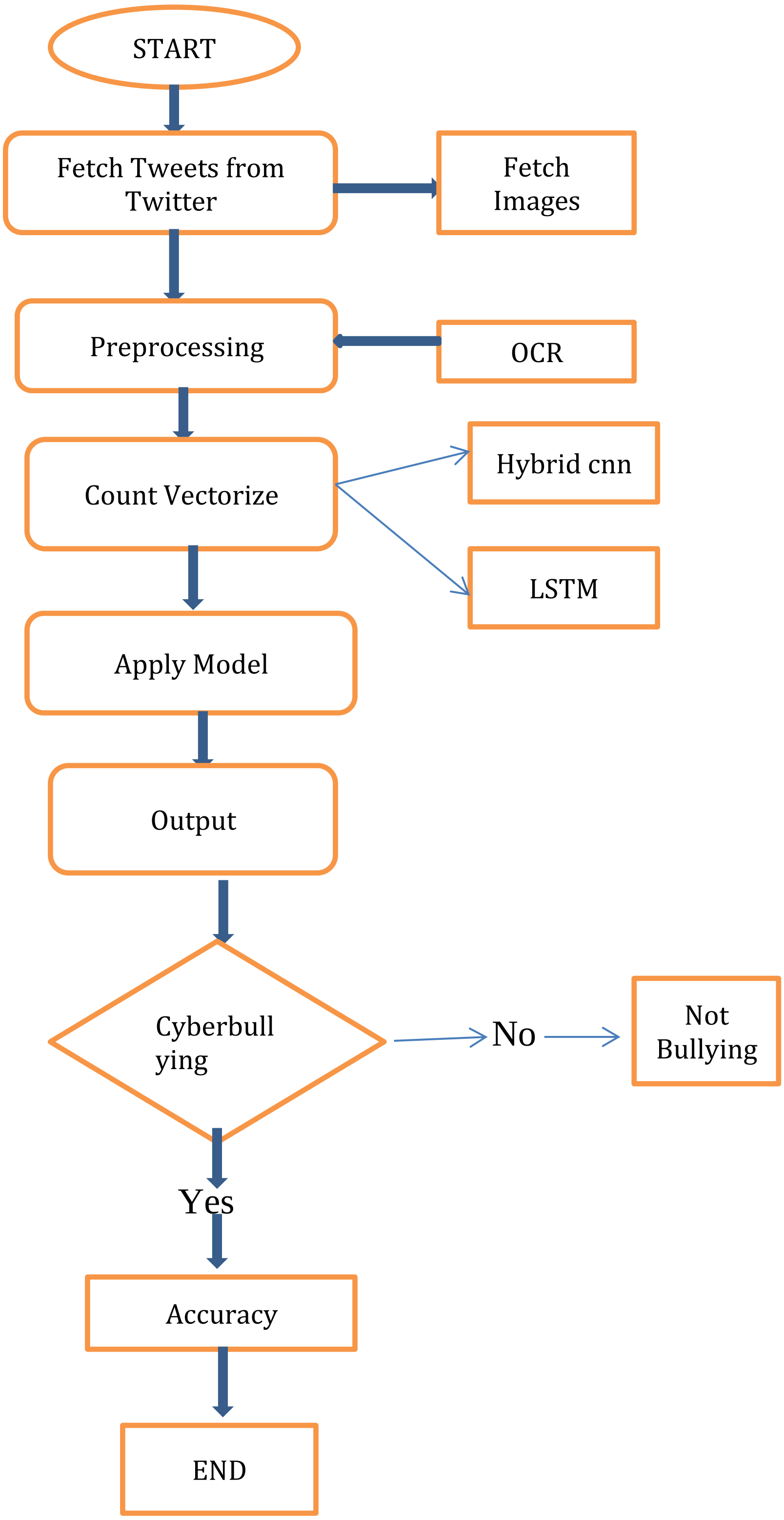
Problem Statement:

The social media network gives us to great communication platform opportunities they also increase the vulnerability of young people to threatening situations online. Cyberbullying on an social media network is a globle phenomenon because of its huge volumes of active users. The trend shows that the cyber bullying on social network is growing rapidly every day. Recent studies report that cyberbullying constitutes a growing problem among youngsters. Successful prevention depends on the adequate detection of potentially harmful messages and the information overload on the Web requires intelligent systems to identify potential risks automatically. So, In this project we focus on to make a model on automatic cyberbullying detection in social media text by modelling posts written by bullies on social network.

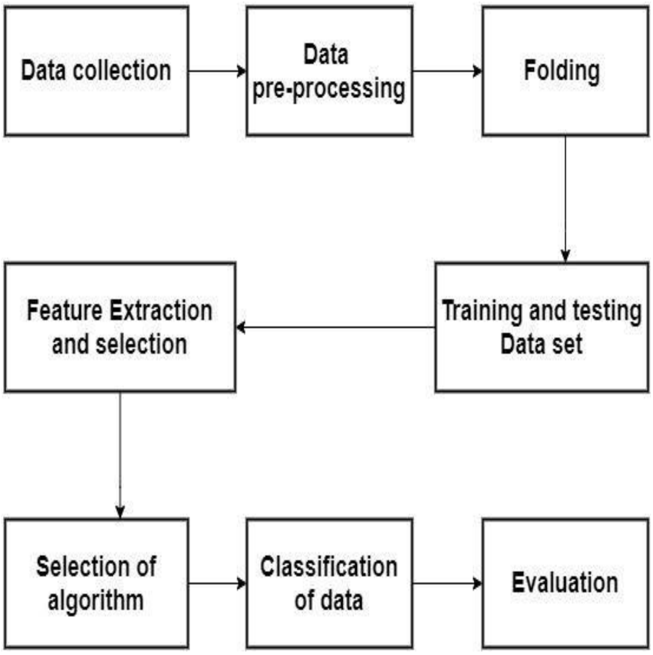
Methodology

This project we will develop using python and web technology. Within that first we will search and find the the dataset and download it for train the model. After downloading first we will pre-process the data and then transferred to Tf-Idf. Then with the help of hybrid cnn, and Lstm,cnn algorithm we train the dataset and generate model separately. Then we are going to develop a web based application using FLASK framework. We will fetch the real time tweets from twitter and then we apply generated model to these fetched tweets and check the text or images are cyberbullying or not.

FLOW CHART:

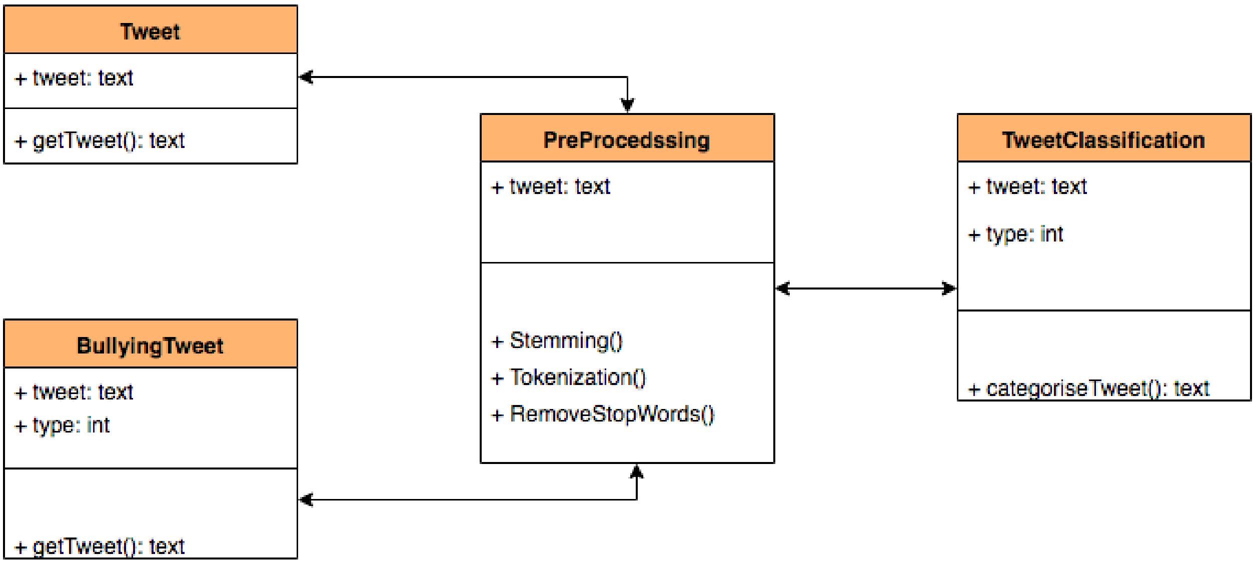

START
Fetch Tweets from Twitter
Fetch Images
Preprocessing
OCR
Count Vectorize
Hybrid cnn
LSTM
Apply Model
Output
Cyberbull ying
No
Not Bullying
Yes
Accuracy
END

ARCHITECTURE DIAGRAM:



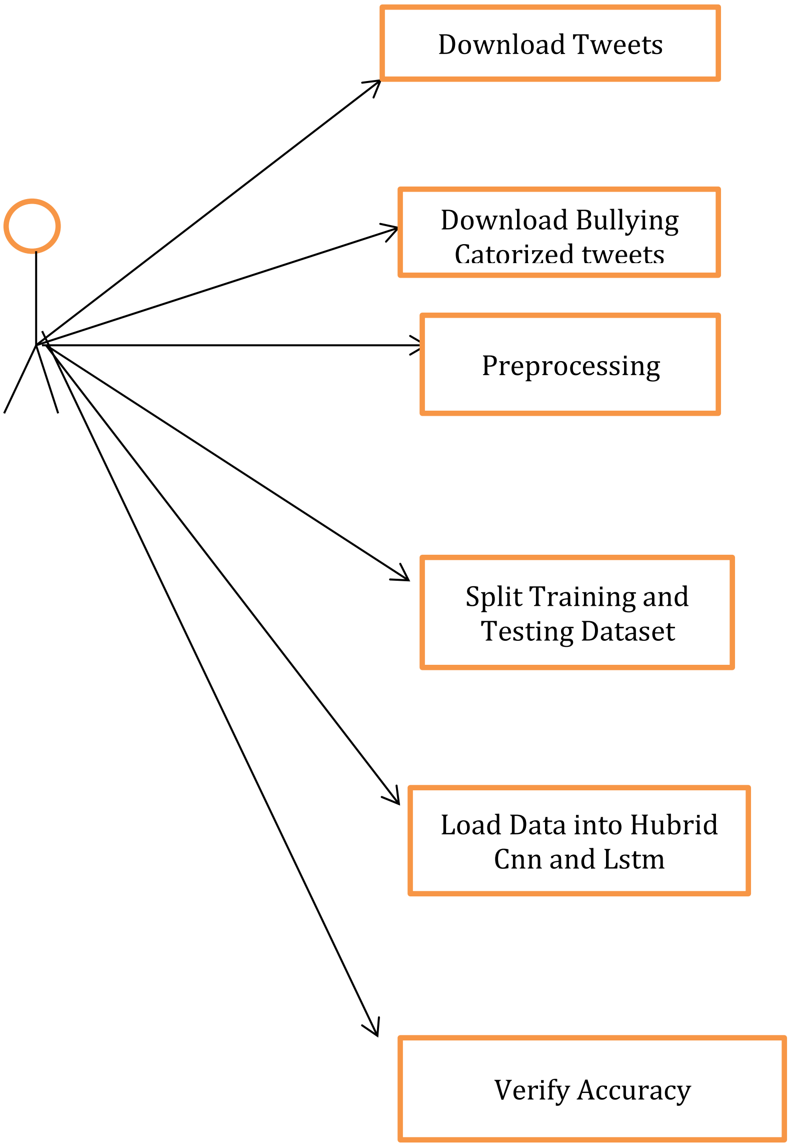
Class Diagram

UML Class diagram shows the static structure of the model. The class diagram is a collection of static modeling elements, such as classes and their relationships, connected as a graph to each other and to their contents.



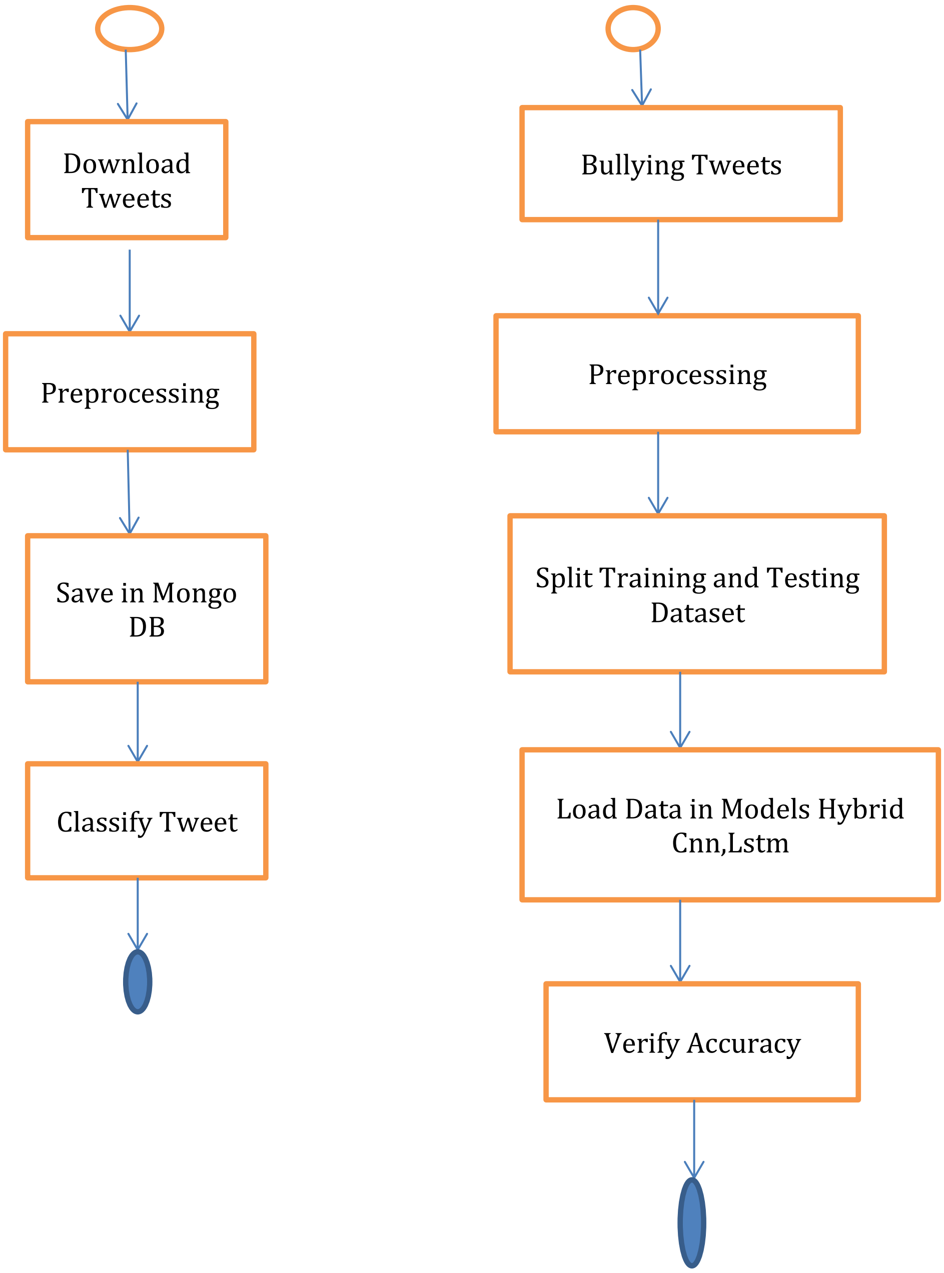
Use Case Diagram

A use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication (participation) associations between the actors and users and generalization among use cases. The use case model defines the outside (actors) and inside (use case) of the system's behavior


Download Tweets
Download Bullying Catorized tweets
Preprocessing
Split Training and Testing Dataset
Load Data into Hubrid Cnn and Lstm
Verify Accuracy

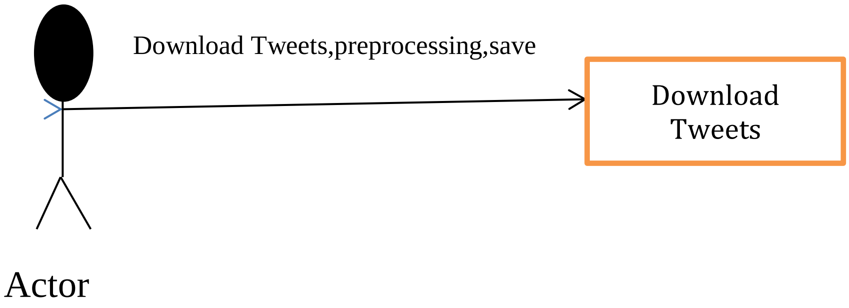
Activity Diagram

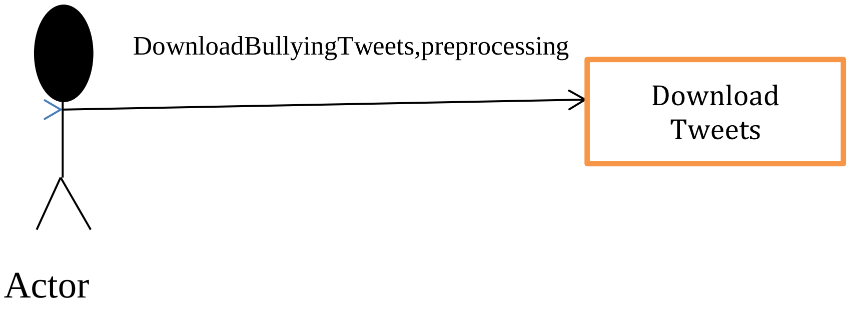
The purpose of activity diagram is to provide a view of flows and what is going on inside a use case or among several classes. Activity diagram can also be used to represent a class's method implementation. A token represents an operation. An activity is shown as a round box containing the name of the operation. An outgoing solid arrow attached to the end of activity symbol indicates a transition triggered by the completion.

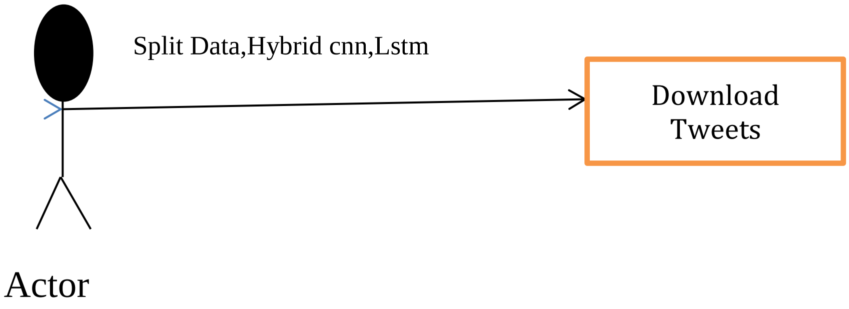


Collaboration Diagram

The collaboration diagram represents a collaboration, which is a set of objects Related in a particular context, and interaction, which is a set of messages exchanged among the objects within the collaboration to achieve a designed Outcome.







The main objective of the proposed system is to detect cyber bullying that occurs on various social media platforms. For the Same we collected the data from different sources such as Twitter, Youtube, Reddit, Wikipedia, etc. The data on these sites is present in the form of tweets and comments from different online users. Next Step is Data preprocessing which means we will process our data before feeding it into our machine. In Data preprocessing, we first remove any irrelevant data from our dataset, then we treat outliers and lastly we handle any missing data which may be present in our dataset. We used 70% of the dataset for training and 30% for testing purpose. A better practice is to use 60% for training, 20% for cross validation, 20% for testing. Due to having such a large dataset both for training and testing, we did not find the necessity to use K-folds cross validation technique. After splitting our dataset intotraining and testing dataset, we will train our machine using this training set which will help our classifier algorithm in learning to classify the data into positive and negative tweets/comments. After the machine has been trained, the testing dataset will be used to test the accuracy of our machine learning model. For evaluation, we calculated the accuracy of the classifier, precision, recall and f- score of the positive, negative and neutral tweets. Precision and recall are the metrics used to determine classifier output quality. Precision is themeasure of how relevant the results are and recall is the measure of how many relevant results are returned. F-score is the average of both the precision and recall.

Software Selection

Python

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and UNIX shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include

Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read − Python code is more clearly defined and visible to the eyes. Easy-to-maintain − Python's source code is fairly easy-to-maintain.

A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

Databases − Python provides interfaces to all major commercial databases.

GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

It supports functional and structured programming methods as well as OOP.

It can be used as a scripting language or can be compiled to byte-code for building large applications.

It provides very high-level dynamic data types and supports dynamic type checking.

It supports automatic garbage collection.

It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Anaconda tool

Anaconda is a FREE enterprise-ready Python distribution for data analytics, processing, and scientific computing. Anaconda comes with Python 2.7 or Python 3.4 and 100+ cross-platforms tested and optimized Python packages. All of the usual Python ecosystem tools work with Anaconda.

Additionally, Anaconda can create custom environments that mix and match different Python versions (2.6, 2.7, 3.3 or 3.4) and other packages into isolated environments and easily switch between them using conda, our innovative multi-platform package manager for Python and other languages.

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS and Linux

Using Python in Anaconda

Many people write Python code using a text editor like Emacs or Vim. Others prefer to use an IDE like Spyder, Wing IDE, PyCharm or Python Tools for Visual Studio. Spyder is a great free IDE that is included with Anaconda. To start Spyder, type the name spyder in a terminal or at the Command Prompt.

The Python 2.7 version of Anaconda also includes a graphical Launcher application that enables you to start IPython Notebook, IPythonQTConsole, and Spyder with a single click. On Mac, double click the Launcher.app, found in your ~/anaconda directory (or wherever you installed Anaconda). On Windows, you'll find Launcher in your Start Menu. The Start Menu also has an Anaconda Command Prompt that, regardless of system and install settings, will launch the Python interpreter installed via Anaconda. This is particularly useful for troubleshooting, if you have multiple Python installations on your system.

CONDA

Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs, runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

Conda as a package manager helps you find and install packages. If you need a package that requires a different version of Python, you do not need to switch to a different environment manager, because conda is also an environment manager. With just a few commands, you can set up a totally separate environment to run that different version of Python, while continuing to run your usual version of Python in your normal environment

Spyder

Spyder's text editor is a multi-language editor with features such as syntax coloring, code analysis (real-time code analysis powered by pyflakes and advanced code analysis using pylint), introspection capabilities such as code completion, calltips and go-to-definition features (powered by rope), function/class browser, horizontal/vertical splitting features, etc.

Spyder is the Scientific Python Development Environment:

Spyder is a powerful interactive development environment for the Python language with advanced editing, interactive testing, debugging and introspection features; and a numerical computing environment thanks to the support of IPython (enhanced interactive Python interpreter) and popular Python libraries such as NumPy (linear algebra), SciPy (signal and image processing) or matplotlib (interactive 2D/3D plotting.

Jupyter notebook

Notebook document

Notebook documents (or "notebooks", all lower case) are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc…). Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc..) as well as executable documents which can be run to perform data analysis.

Jupyter Notebook App

The Jupyter Notebook App is a server-client application that allows editing and running notebook documents via a web browser. The Jupyter Notebook App can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet.

In addition to displaying/editing/running notebook documents, the Jupyter Notebook App has a "Dashboard" (Notebook Dashboard), a "control panel" showing local files and allowing to open notebook documents or shutting down their kernels.

Kernel

A notebook kernel is a "computational engine" that executes the code contained in a Notebook document. The ipython kernel, referenced in this guide, executes python code. Kernels for many other languages exist (official kernels).

When you open a Notebook document, the associated kernel is automatically launched. When the notebook is executed (either cell-by-cell or with menu Cell -> Run All), the kernel performs the computation and produces the results. Depending on the type of computations, the kernel may consume significant CPU and RAM. Note that the RAM is not released until the kernel is shut-down

Notebook Dashboard

The Notebook Dashboard is the component which is shown first when you launch Jupyter Notebook App. The Notebook Dashboard is mainly used to open notebook documents, and to manage the running kernels (visualize and shutdown).

The Notebook Dashboard has other features similar to a file manager, namely navigating folders and renaming/deleting files.

In this case, "notebook" or "notebook documents" denote documents that contain both code and rich text elements, such as figures, links, and equations. Because of the mix of code and text elements, these documents are the ideal place to bring together an analysis description and its results as well as they can be executed perform the data analysis in real time.

These documents are produced by the Jupyter Notebook App.

For now, you should just know that "Jupyter" is a loose acronym meaning Julia, Python, and R. These programming languages were the first target languages of the Jupyter application, but nowadays, the notebook technology also supports many other languages.

Requirement Specifications

A Software Requirements Specification (SRS) is a description of a particular software product, program or set of programs that performs a set of functions in a target environment.

Overview

Social Media allows the creation and exchange of user-generated content. Via social media, people can enjoy enormous information, convenient communication experience and so on. However, social media may have some side effects such as cyber bullying, which may have negative impacts on the life of people, especially children and teenagers. Cyber bullying can be defined as aggressive, intentional actions performed by an individual or a group of people via digital communication methods such as sending messages and posting comments against a victim. As a side effect of increasingly popular social media, cyber bullying has emerged as a serious problem afflicting children, adolescents and young adults. Machine learning techniques make automatic detection of bullying messages in social media possible, and this could help to construct a healthy and safe social media environment.

The SRS contains the details of process, functions of the product, user characteristics. The non-functional requirements if any are also specified.

Scope and Purpose

The purpose of software requirements specification specifies the intentions and intended audience of the SRS. The scope of the SRS identifies the software product to be produced, the capabilities, application, relevant objects etc.

Product Description

Product Perspective states if product is self-contained, independent or if product is part of a large system. This system implements an algorithm consists of three phases. Product Function describes major function the software will perform. Product functions should be organized so that they are understandable to the client or anyone else who read SRS for the first time. User Characteristics indicates intended users of product and education level, experience, technical expertise required by user. This application implements various strategies for execution of the code on various data sets. Since the system is implemented in Python, the request is initiated by any data source in the form of a request at the application.

Specific Requirements

This section of the SRS provides a description of the observable behavior of a software system. Since we deploy Machine Learning to identify pattern analysis of data, we use Python as programming Language. Python has support of numerous machine learning libraries that can be tested easily and quickly. Also we have preferred Jupyter Notebook as IDE as this is very interactive and provides output very easily for our quick observation.

Functional Requirements

Functional requirements will define the fundamental actions that must take place in the software in accepting & processing the inputs in processing & generating the outputs. Class diagram, Uses Case Diagram, Activity Diagrams, Sequence Diagrams, and Collaboration Diagrams will be provided which describes the flow of data between various processes of the system. Process descriptions will be provided based on the process information. Use Case

Specification will be enclosed and provided which describes the detailed specifications of each use case.

The data must be processed from different data sources in a finite amount of time.

Non- Functional Requirements

The major non-functional Requirements of the system are as follows

1. Usability

The system is designed with completely automated process hence there is no or less user intervention.

2. Reliability

The system is more reliable because of the qualities that are inherited from the chosen platform python. The code built by using python and is more reliable.

3. Performance

This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

4. Supportability

The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is having python installed in the system.

5. Implementation

The system is implemented with Python environment. We use Anaconda and Jupyter Notebook for development and testing.

Software system attributes

Scalability: The number of intermediate sources can be scalable, thus changing or updating the data.

Reliability: This proposed system should provide reliable results.

Resource Utilization Efficiency: This system will utilize less processing time.

Security: This system is developed in java hence it is secured

Safety: This system uses java code safety

Capacity: Any number of users will be able to use this system

Interfaces: They will be provided in the design document

Availability: This system will always cater to the needs of the users
Accuracy: This system will produce accurate results

Reusability: This system can be easily reused

Ease of Use: This system is developed using graphical user interface hence it is easy to use

Interoperability: Through the use of Preprocessed file interoperability is achieved 24

Portability: The system is portability any version of windows as well as Linux Systems.

Privacy: This system ensures privacy of the data

System Administration Ease: This system will provide easy administration capabilities

Expandability: Any number of modules can be added to this system

Maintainability: The system would be design as open system and new method is easily added

Testability: Test cases will be written to ensure correct results

PROJECT DESCRIPTION

Problem Statement

Feature selection techniques are commonly utilized as a preprocessing stage for clustering, in order to overcome the curse of dimensionality. The most informative dimensions are selected by eliminating irrelevant and redundant ones. Such techniques speed up clustering algorithms and improve their performance. Nevertheless, in some applications, different clusters may exist in different subspaces spanned by different dimensions. In such cases, dimension reduction using a conventional feature selection technique may lead to substantial information loss.

Algorithm Analysis

List of Modules

1.Download Tweets

2. Preprocessing

3.Training and testing split

4.Hybrid cnn and lstm

5.Test and validation

6.Oppression detection

Module Description

Download Tweets

Twitter is a popular social network where users share messages called tweets. Twitter allows us to mine the data of any user using Twitter API or Tweepy. The data will be tweets extracted from the user. The first thing to do is get the consumer key, consumer secret, access key and access secret from twitter developer available easily for each user. These keys will help the API for authentication

Steps to obtain keys:

- Login to twitter developer section Go to “Create an App”
- Fill the details of the application.
- Click on Create your Twitter Application
- Details of your new app will be shown along with consumer key and consumer secret.
- For access token, click ” Create my access token”. The page will refresh and generate access token.

Tweepy is one of the library that should be installed using pip. Now in order to authorize our app to access Twitter on our behalf, we need to use the OAuth Interface. Tweepy provides the convenient Cursor interface to iterate through different types of objects. Twitter allows a maximum of 3200 tweets for extraction.

Oppression Tweets Dataset

With the increasing use of social media, cyber bullying behavior has received more and more attention. Cyber bullying may cause many serious and negative impacts on a person's life and even lead to teen suicide. To reduce and stop cyber bullying, one effective solution is to automatically detect bullying content based on appropriate machine learning and natural language processing techniques.

Basic requirement to apply machine learning algorithms and test data mining is collecting respective dataset. We need appropriately categorized data where each tweet is categorized as bullying or not. To make apply and test different algorithms for these kind of projects, datasets are created so that students and researchers can apply their algorithms on these dataset and validate results.

PreProcessing

Data Preprocessing improves the data set so that the dataset includes only required information. The various stages in data preprocessing are

Tokenization

Tokenization is the process of distributing large set of unstructured messages into a small subset of tokens. These are classified with the help of various aspects such as white spaces, punctuation marks and is categorized as phrases, sentences etc

Stop Word Removal

The most common words that are used in a text are words such as 'a' 'and' 'are' and so on. The main drawback of these words are that such words only contribute very little meaning to text and aids only a very small value in classifying text. Stop words removal from messages results in more convenient recognition of text in further steps.

Replacement of Special Characters

The method deals with the replacement of special characters like '@' with its exact word ‘at’. In tweets, this step has larger importance because of the extensive occurrences of special symbols.

Stemming and Lemmatization

This method finds the root of a single word and is considered as a heuristic technique it simply abridges prefixes as well as suffixes. It uses word-based approach in order and is so called dictionary based approach. Lemmatization method can be considered as the further extension of stemming technology. For the grammatical categorization of characters to get the base method of a single word called lemma, this technique is widely used. One of the algorithms which are broadly used for this purpose is Porters algorithm and can be more adequately used and is more specific.

Training and Testing Split

Pre-processed data needs to be feed into appropriate models for analyzing. Existing methods used Naïve Bayes algorithm for tweet classification. In our study, we will be categorizing tweets as bullying or not using below algorthims

Hybrid cnn and Lstm:

A CNN-LSTM model is a combination of CNN layers that extract the feature from input data and LSTMs layers to provide sequence prediction. The CNN-LSTM is generally used for activity recognition, image labeling, and video labeling. Their common features are that they are developed for the application of visual time series prediction problems and generating textual annotations from image sequences.

REFERENCES

1.Poeter.(2011)Study: A Quarter of Parents Say Their Child Involved in Cyberbullying. pcmag.com. [Online].Available: http://www.pcmag.com/article2/0,2817,2388540,00.asp

2.J. W. Patchin and S. Hinduja, "Bullies move Beyond the Schoolyard; a Preliminary Look at Cyberbullying," Youth Violence and Juvenile Justice, vol. 4, no. 2, pp. 148–169,2006

3. Anti-Defamation League. (2011) Glossary of Cyber bullying Terms.adl.org.[Online].Available:http://www.adl.org/education/curriculum connections/cyber bullying /glossary.pdf

4.N. E. Willard, Cyber bullying and Cyber threats: Responding to the Challenge of Online Social Aggression, Threats, and Distress. Research Press, 2007.

5.D. Maher, "Cyberbullying: an Ethnographic Case Study of one Australian Upper Primary School Class," Youth Studies Australia, vol. 27, no. 4, pp. 50–57, 2008.

https://www.sciencedirect.com/topics/computer-science/deep-neural-network

CONCLUSION:

The goal of this project is to the automatic detection of cyberbullying-related posts on social media. Given the information overload on the web, manual monitoring for cyberbullying has become unfeasible. Automatic detection of signals of cyberbullying would enhance moderation and allow to respond quickly when necessary. However, these posts could just as well indicate that cyberbullying is going on. The main aim of this project is that it presents a system to automatically detect signals of cyberbullying on social media, including different types of cyberbullying, covering posts from bullies, victims and bystanders.

FUTURE SCOPE:

In future author suggests that the developed prototype can be implemented on Wide Area Network with combination of different dictionaries. Cloud Frameworks can be used for this purpose. The fuzzy rules can be further modified to detect non textual forms of cyberbullying such as abuse through obstructed texts, images and videos.