International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 04 | Apr 2025 www.irjet.net p-ISSN: 2395-0072

Analysis of Procedures for Detecting Malicious URLs and Classifying Community Posts on Social Media Websites

Ritwik Sinha[1], Rahul Gupta[2]

[1]M.Tech. (CSE) Scholar, Department of Computer Science and Engineering, S. R. Institute of Management and Technology Lucknow, Uttar Pradesh, India
[2]Assistant Professor, Department of Computer Science and Engineering, S. R. Institute of Management and Technology Lucknow, Uttar Pradesh, India

---

Abstract - Millions of microblogs and social media websites post public sentiments and ideas alongside information about the present situation following crises or tragedy events. While the majority of prior research has focused on gathering contextual information, it focusses on a particular non-situational category of tweets—that is, social tweets—that make derogatory remarks targeting other racial or religious groups. To develop a classifier that performs significantly better than current methods for separating tweets that are communal from those that are not. Paradoxically, a significant portion of group tweets are posted by well-known people, the majority of whom discuss politics and the media. Furthermore, those who tweet about communities build robust, social media-related communities. Therefore, for security purposes, it is necessary to categorise such posts from websites. This thesis proposes a novel, non-event driven strategy for characterising communal microblogs that will be useful in the event of a calamity. Additionally, a process is designed to identify dangerous URLs in these kinds of situations for security reasons.

Key Words: media websites post, fake tweets, social media, communal microblogs etc .

1. INTRODUCTION

Online social media (OSM) such as Twitter and Facebook are really plagued by hostile and depressing substances like trolling, digital bullying, hate speech, etc. today. Different types of hostile substances. Hate speech can be divided into a few classes in which people address different characteristics, e.g., religion, sexual orientation, gender, ethnicity, nationality, etc., of the objective meeting [6]. Kinds of contemptuous speeches, in this work they focus on a particularly hurtful and potentially dangerous class: communal tweets coordinated with certain strict or racial networks, for example "Hindu","Muslim", "Christian" etc. In particular, study communal tweets posted during times of disaster or crisis. A fiasco circumstance largely affects the minds of the majority and renders them powerless. Under such circumstances, contempt and falsehood grow in the affected area, which can lead to a real weakening of peace relations. In this thesis, present a point-by-point investigation of community tweets published under catastrophic circumstances, e.g., programmed recognition of such tweets who break down customers who post such tweets and also recommend an approach to counteract this substance. These hostile tweets were previously posted frequently during man-made disasters such as militant psychological attacks. For example, Burnap and Williams have thrown light on U. During the Woolwich raid, K .masses focused on a specific strict network that the raiders are associated with events such as floods and earthquakes. Some examples of community tweets are given in Table 1.1.

Table 1.1: Tweets Set

SR. No.	Text
1	F**k these *Missionaries* who are scavenging from whatever's left after the #Nepal Earth quake Have some shame & humanity.
2	Dear #kashmir Floods take away all rapist *muhammad's piglets* out of kashmir with you, who forced out kashmiri *Hindus* from their motherland!!
3	*Radical Muslims* want to behead u, moderate Muslims want radical Muslims to behead you n liberals want to save them. result. #Gurdaspur Attack.
4	RT @polly: # Hillary Clinton's reply when asked if war on terror is a war on *"radical Islam"* #Dem Debate.
5	Jesus F***ing Christ ... Active shooter reported in San Bernardino, California.

Social media platforms' explosive expansion has completely changed how people communicate, exchange information, and build communities. Social media platforms like Facebook, Instagram, Reddit, and Twitter (now X) have become vital information channels for anything from breaking news and personal updates to political debate and business advertisements. The dissemination of malicious URLs is one of the most common ways that malicious groups try to take advantage of consumers, yet this widespread influence also draws them in.

1.1. Objective of the Research

The objectives of this thesis are:

- To do extensive research works on detection of hate or loathing speech over social media websites.
- To implement a classifier on detection of communal posts from data set of tweets.
- To modify the features in such a way to detect the malicious user's URL tweets in the post's dataset.
- To propose a method to detect both malicious URL and detection and classification of communal and non-communal posts.

2. LITERATURE REVIEW

2.1. Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modelling for Policy and Decision Making

The application of "big data" to focus and fundamental leadership is a hot topic right now. The murder of drummer Lee Rigby in Woolwich, London, UK, in 2013 garnered extensive social media attention and offered a chance to consider the proliferation of digital hate on Twitter. In order to create and test a controlled AI content classifier that would recognise teasing or possibly hostile replies connected to race, ethnicity, or religion, as well as larger and wider responses, one person wrote on Twitter that the data was gathered shortly after Rigby's death. The key components of characterisation, which include the syntactic situations between phrases to perceive "other" phrases, the encouragement to respond with the opposite activity, and instances of established or defended social gatherings of oppression, have been taken from the content of each tweet. They performed best when they combined probabilistic, rule-based, and spatial classifiers with a metaclassifier to gather votes. The author demonstrates how the classifier's aftereffects may be efficiently used in a measurable model to assume the plausible spread of online hate using data from Twitter. It concerns mediation applications and fundamental leadership. [1]

In this article, the author created a managed AI classifier for despicable and negative substances on Twitter. The classifier's goal is to assist managers and authorities in observing the public response to emotional events on a broad scale, such as the 2013 Woolwich murder of drummer Lee Rigby. According to earlier studies, 58 percent of negative acts that took place after September 11th were hated two weeks after the incident (4 percent of the risk period). The information is continuously accessible through online social networks and microblogging platforms like Twitter, which can help us weed out the pervasive hate and hostile responses on the internet as soon as the threat of hateful responses is greatest.

2.2. Analysing the Targets of Hate in Online social media

Social media frameworks provide internet customers with a harmonious frame work in which to express their thoughts and assessments uninhibited. While this quality speaks of incredible and extraordinary mailings, it also brings with it significant difficulties. One unique instance of this kind of issue is hate speech on the internet. The concept of hate speech on social media is not well understood, despite its magnitude and scope. The author uses Twitter and Whisper as two social media platforms to do this. At this stage, the author develops and authorises a method for differentiating between hate speech in both frameworks. By providing the avoidance and recognition of raw titles close by, the author's findings differentiate Internet contemptuous speech structures and provide a more comprehensive comprehension of the issue.

A increasing number of influential people and dynamic associations, as well as governments and commercial companies, are joining the fight against the blatant rhetoric of disgust on the Internet. This important problem in avant-garde society is now evident. The author's efforts even reveal new types of loathing on the internet that are not really bad deeds but can be dangerous to people. The author is confident that the author's data set and methodology can help review the frames and location calculations to spot novel slogans identified with hate speech as well as moving increasingly exposed components to shift language of hate on the internet distinguish. Research plan.[3]

3. CHARACTERIZING OF TWEETS FROM MICROBLOGS

The system improvement lifecycle (SDLC) or software program improvement lifecycle in framework layout, facts frameworks, and programming layout is the manner to create or adjust frameworks and fashions and approaches that human beings use to create the ones frameworks. The SDLC concept helps many sorts of programming improvement systems. These philosophies shape the device that allows you to prepare and manipulate the introduction of a facts body with inside the product improvement process.

3.1. PROPOSED WORK

3.1.1 Methodology

The author for the current communal post detection system has provided a concept for detecting regular or communal tweets but has not provided a concept for detecting malicious users, and malicious users are often

responsible for spreading communal tweets, so a new proposed concept is applied to detect such malicious users. Almost every tweet contains a URL link to a video or other greeting document. Since URLs can have a long text, but Twitter will only support 140 characters, Twitter has adopted the SHORTEN URL rule, in which a long URL can be mapped to a shorter URL. Shorten URLs may be used in tweets, and when a user clicks on one, Twitter obtains a large URL mapping from Shorten URL.

Example

Stack overflows shorten URL

Short URL = http://s.tk/

When one paste above short URL in browser and then press enter key then automatically that URL changes to big URL as below one

Big URL = https://stackexchange.com/

Malicious users could use this strategy to disperse tweets with Shorten URLs that will redirect users to malicious websites as they click on them. The malicious website then extracts data from the user's computer and sends it to other malicious users.

Malicious users will always have only one or a few websites, and they will build thousands and thousands of shorten URLs that point to those few malicious websites. Users can be routed to such web pages if they click on such URLs.

Twitter already uses blacklisted URLs to detect suspicious connections, but this is insufficient to detect various shorten URLs. To fix this challenge, all URLs can be investigated to see if the same webpage is being redirected by each of them. A URL may be flagged as dangerous if multiple Shorten URLS drive users to the same website; maintaining a blacklist of URLs is not necessary. This is the system's suggested concept feature technique.

Below is the code to get expand URL from short URL

public static String expand URL (String shortened URL) throws IO Exception.

3.1.2 Unified Modeling Language diagrams

A software engineer can use modelling notation, which is determined by a set of pragmatic and semantic syntactic rules, to define an analytical model with the UML. Five distinct viewpoints, each describing the system from a different angle, are used to illustrate a UML system. Each view is defined by a series of diagrams that are shown below.

View of the User Model

i. The system is rendered from the user's point of view in this view.
ii. The analysis's rendering provides an end-user perspective on a usage scenario.

View of the Structure Model

i. This paradigm is used to retrieve the system's data and capabilities.
ii. This model view represents the static structures.

The behaviour model view depicts the interactions between different structural elements as well as behavioural dynamics as system components, as shown in the user model and structural model view.

Implementation-Model View:

This shows how the system's behavioural and structural elements will be developed.

Environmental-Model View:

This shows the system to be used as well as the behavioural and structural features of the environment.

3.1.3. Diagram of the class

The most crucial component of object-oriented modelling is the class diagram. Both technical modelling, which converts concepts into programming code, and wide conceptual modelling of software systems are done with it. Facts can also be modelled using class diagrams. Along with the interactions inside the software and the training that needs to be coded, they are the main components of a category diagram.

A magnificence with 3 sections, within side the diagram the training are represented via way of means of packing containers with 3 parts:

- The higher element bears the call of the magnificence.
- The centre element consists of the attributes.
- The decrease element consists of the techniques or operations that carry out or be capable of carry out the magnificence

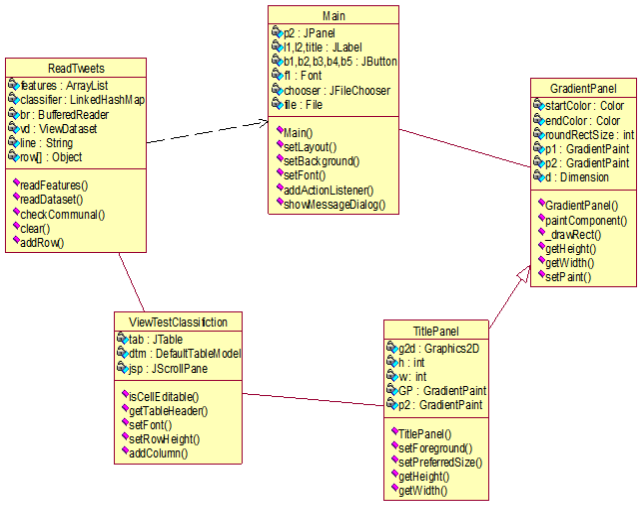

Figure 3.1: Class diagram

3.1.4 Use case diagram

In its simplest form, a use case diagram is a representation of a user's interaction with a system that outlines the requirements of a use case. The many types of system users and the different ways they can communicate with the system are shown in a use case diagram. Other diagram kinds usually follow this one, which is usually used in conjunction with a written use case.

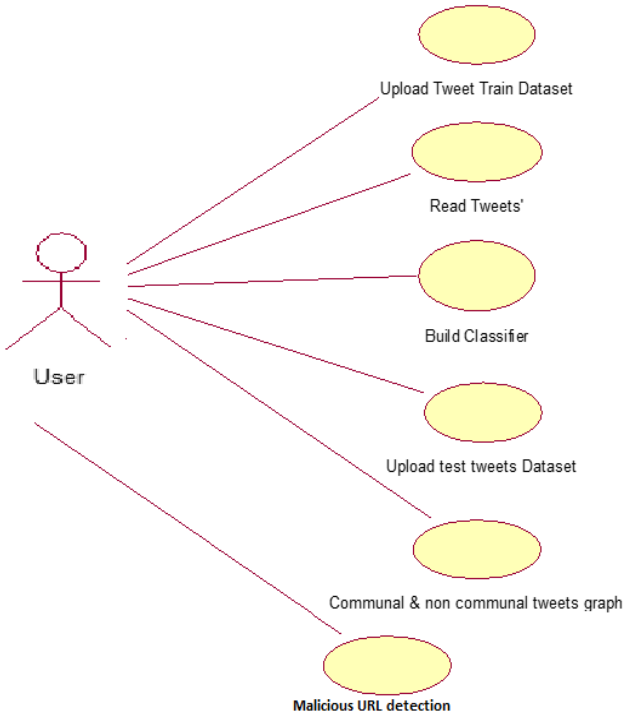

Figure 3.2 Use case diagram

3.1.5 Activity diagram

Another helpful UML diagram for illustrating intricate system components is the activity diagram. In essence, it is a flowchart that shows how data moves between processes. The activity can be specified by a machine process. Control is consequently passed from one action to the next. This flow may naturally be synchronous, branched, or sequential.

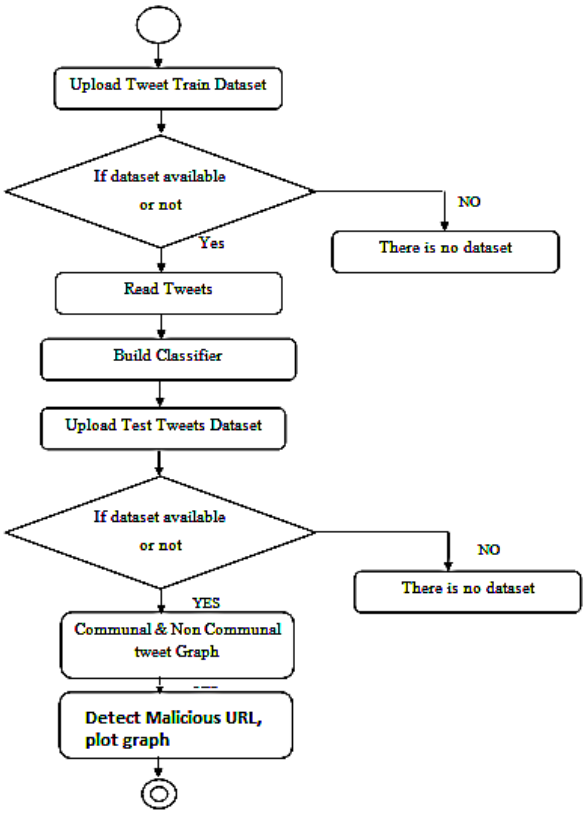

Figure 3.3: Activity diagram

3.1.6 Data Flow Diagram

DF diagrams display the inputs and outputs of a system's data processing. Data flow diagrams are a useful tool for successfully illustrating any business function. The method starts with a general summary of the business and progresses through each of them. the functionally relevant areas. The amount of depth required for this analysis can be achieved. The methodology uses a technique called top-down expansion to complete the analysis.

As the name suggests, a data flow diagram (DFD) is a graphic representation of the information movement within a process. A DFD can be made with basic symbols. Furthermore, simple, freely downloadable graphical tools for creating DFDs make it simple to automate complex processes. One paradigm for creating and assessing

information processes is a DFD. DFD uses inputs and outputs to show how information moves through a process. A DFD is also known as a process model. A DFD illustrates a technological or business process supported by stored external data, data flowing between processes, and the outcomes.

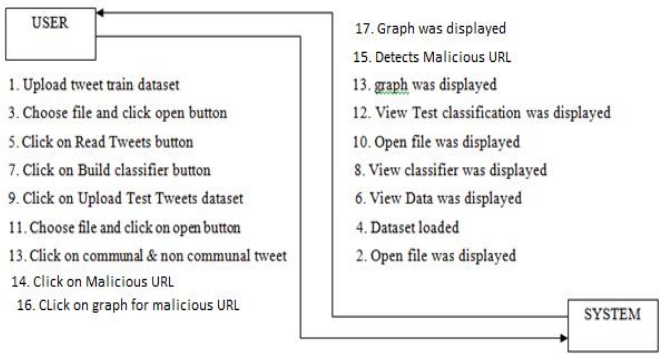

Figure 3.4: Data Flow Diagram

4. TESTING AND RESULTS

4.1 TESTING

4.1.1 Testing and Deployment:

One of the most vital activities in an undertaking is deployment or implementation. This is the method wherein one needs to be vigilant, ensuring that all efforts made throughout the undertaking are interactive. The most crucial step in achieving a stable and strong system, while assuring clients that the implementation of the new system is dependable and usable. During implementation, each application is independently tested using sample data to make sure it works as required by the program specifications. To guarantee user comfort, the operational tool and its surroundings have been thoroughly examined.

4.1.1.1 Deployment

The implementation stage is less creative than the device design stage. Its main objectives are file conversion and user education. A lot of user training might be necessary for the device. The system's starting parameters should be modified appropriately as a result of programming. To help the user comprehend the many functions, a fundamental working technique is presented. Depending on the user's access, reports can be printed using either a dot matrix printer or an inkjet printer. Setting up the suggested framework is easy.

Generally speaking, "implementation" describes the procedure that turns a revised or new device specification into a working system.

4.1.1.2 Testing

The process of creating test data and using it to look at specific modules and the validation offered for the fields is called testing. Device testing is the following stage, which confirms that every part of the system functions as a whole. It is important to select the test data such that it can be used to a range of scenarios. In actuality, testing is the stage of development where the device is examined to make sure it functions correctly and consistently before the actual process starts. A list of the research methods employed during the testing procedure is provided below.

4.1.2 System Testing

In the information technology sector, testing is now a crucial component of all projects and products. the significance of testing in establishing whether one should resist or whether one is prepared to move on. It is impossible to overstate how serious a situation is, which is why pre-production testing is essential: Software should be tested to make sure it is performing the intended function before being made available to the general public. Several methods will be used in the study to confirm the software's reliability. A program execution sequence for a data series was simulated, and the program was logically evaluated. Consequently, the Code has been carefully examined for any potential flaws or recommendations.

4.1.3 Module Testing

Errors are found by individually verifying each module. This enables us to find and fix errors without interfering with other modules. To achieve the intended result, a program must be fixed if it is unable to complete the necessary operation. Separate grades are assigned to each module, starting with the smallest and lowest and working up to the next level. Every gadget module is examined independently. The job classification module, for instance, is examined independently. The average execution of many jobs is used to verify this module. The outcomes are contrasted with those that were manually produced. The findings imply that the suggested system outperforms the existing one in terms of effectiveness. Every gadget module is examined independently. Job planning and resource classification modules are assessed independently in the process's waiting time is shortened by this plan and the associated outcomes that are gathered.

4.1.4 Integration Testing

Integration testing is done in addition to module inspection. When connecting the modules, there is a chance that faults will arise; this inspection will fix these mistakes. In this framework, both modules are connected and verified. Consequently, the system completes the resource job assignment successfully.

4.1.5 Acceptance Testing

When the user finds no significant problems with the accuracy of the system, it passes a final acceptance test. Without needing actual implementation, this test ensures that the framework satisfies the initial goals, objectives, and requirements set during the review, saving administrators and users time and money. Acceptance test Finally, the time has come, and you are prepared to depart.

Table 4.1: Test cases

Test Case Id	Test Case Name	Test Case Desc	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
01	Upload Tweet Train Dataset	Verify Whether dataset uploaded or not	If not upload	cannot get results	dataset loaded	High	High
02	Read Tweets	Check test tweets is reading or not	If it's not reading	cannot Get dataset	View Dataset displayed	High	High
03	Build Classifier	Verify whether classifier is processing or not	If it's not processing	cannot get classifier	View Classifier displayed	High	High
04	Upload test tweets dataset	Verify whether dataset is availablle or not	If it's not availablle	cannot see test classification	View Test classification	High	High
05	Communal & Non communal Tweets Graph	Verify the graph is processing or not	If it's not processing	cannot get tweet graph	Communal & non communal graph displayed	High	High

4.2 RESULTS & SCREENSHOTS

4.2.1 Classification of Micro blogs & detection of Malicious URL proposed work results:

Results and screenshots of the current system implementation are displayed in this section. Figure 4.1 displays the application GUI, or graphical user interface, window.

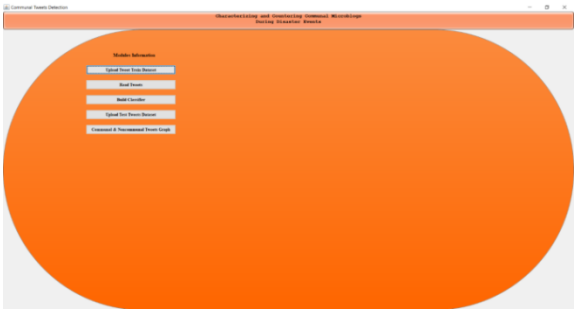

Figure 4.1: GUI of implementation of existing system

The buttons on the GUI are: Read Tweets, Compile Classifier, Load Tweet Dataset for Testing, Load Tweet Stream Dataset, and Results Diagram for Communal and Non-Communal Results.

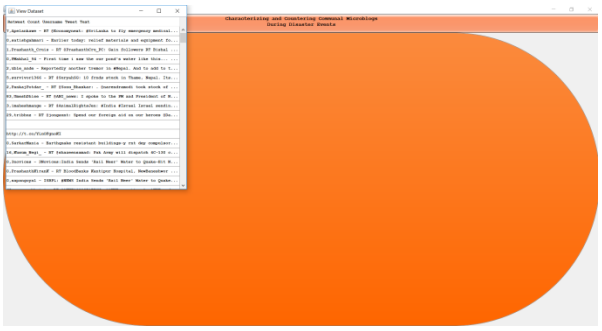

Figure 4.2: Dataset for tweets

The data set of trained tweets is displayed in figure 4.2. Classification following the build classifier button is displayed in figure 4.3.

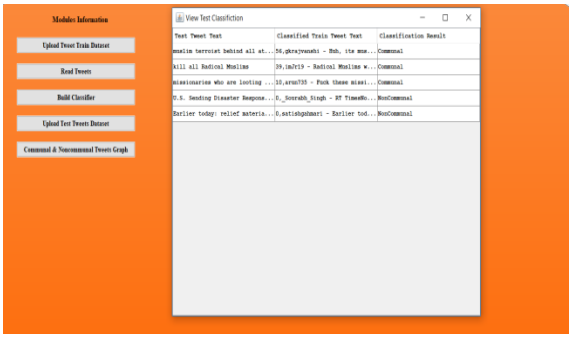

Figure 4.4: Test Results of Classification

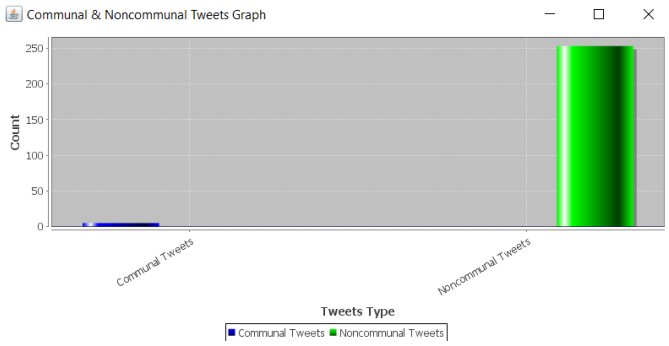

Figure 4.5: Communal & Non-communal tweets graph

In figure 4.4, test results after classification are shown. In figure 4.5, the chart of communal and non-communal tweets is shown

5. CONCLUSIONS AND FUTURE SCOPE

5.1. Conclusions

This thesis is the main project aimed at characterising group tweets that were posted during the disaster and looking into the users who posted those tweets with malicious URLs. An event-independent classifier is suggested here, which might be used to sort through shared tweets beforehand. Additionally, it was found that public tweets are heavily retweeted and shared with the help of a number of well-known users; they typically contain information about media and governmental issues. Clients who use and advance communal substances have a strong social tie with one another. Additionally, the vast majority of patrons unexpectedly explode as a result of these kinds of actions, particularly their animosity towards the event's expressly stringent networks. It's important to reduce the potential negative effects of communal tweets since, in times of crisis, some consumers frequently submit to communal content, which annoys people. have suggested an event-independent classifier to distinguish between these hostile tweets. However, it has been discovered that these anti-community tweets are retweeted far less frequently than those that are communal, and they may also no longer be as widely shared. Finally, a continuous framework that classifies tweets as either communal or non-communal and diagnoses malicious URLs for security purposes.

5.2. Future Scope

There are numerous ways in which this research can be enhanced in the future. Advanced AI and language processing techniques can be used to increase the accuracy of the event-independent classifier. Real-time technologies that collaborate closely with social media platforms can be created to identify and stop the propagation of damaging community tweets. Scams, false information, and cyber threats can all be detected by expanding the identification of harmful links in tweets. One way to understand why people post such stuff on social media and how to prevent it is to look at how they respond to crises. To observe how community content travels across many networks, this study can potentially be expanded to other social media sites like Facebook and Instagram. These findings can be used by policymakers to develop more effective regulations that protect free expression while limiting harmful information. Lastly, social media may be made safer in emergency situations by developing scalable AI systems that can manage massive volumes of data and automatically get better over time.

REFERENCES

[1] P. Burnap and M. L. Williams, “Cyber hate speech on Twitter: An application of machine classification and statistical modeling for policy and decision making,” Policy Internet, vol. 7, no. 2, pp. 223–242, 2015.

[2] I. Chaudhry, “#Hashtagging hate: Using Twitter to track racism online,” First Monday, vol. 20, no. 2, 2015. [Online]. Available: http://firstmonday.org/ojs/index.php/fm/article/view/5450

[3] L. A. Silva, M. Mondal, D. Correa, F. Benevenuto, and I. Weber, “Analyzing the targets of hate in online social media,” in Proc. ICWSM, Mar. 2016, pp. 687–690.

[4] N. D. Gitari, Z. Zuping, H. Damien, and J. Long, “A lexicon-based approach for hate speech detection,” Int. J. Multimedia Ubiquitous Eng., vol. 10, no. 4, pp. 215–230, 2015.

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

[6] M. Mondal, L. A. Silva, and F. Benevenuto, “A measurement study of hate speech in social media,” in Proc. ACM HT, 2017, pp. 85–94.

[7] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic, and N. Bhamidipati, “Hate speech detection with comment embeddings,” in Proc. WWW, 2015, pp. 29–30.

[8] W. Magdy, K. Darwish, N. Abokhodair, A. Rahimi, and T. Baldwin, “#ISISisNotIslam or #DeportAllMuslims?: Predicting unspoken views,” in Proc. ACM Web Sci., 2016, pp. 95–106.

[9] K. Rudra, A. Sharma, N. Ganguly, and S. Ghosh, “Characterizing communal microblogs during disaster events,” in Proc. IEEE/ACM ASONAM, Aug. 2016, pp. 96–99.

[10] E. Greevy and A. F. Smeaton, "Classifying racist texts using a support vector machine," in Proc. SIGIR, 2004, pp. 468–469.

[11] N. Pendar, "Toward spotting the pedophile telling victim from predator in text chats," in Proc. ICSC, Sep. 2007, pp. 235–241.

[12] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in Proc. Int. Conf. Social Comput. Privacy, Secur., Risk Trust (PASSAT), (SocialCom), Sep. 2012, pp. 71–80.

[13] K. Dinakar, B. Jones, C. Havasi, H. Lieberman, and R. Picard, "Common sense reasoning for detection, prevention, and mitigation of cyberbullying," ACM Trans. Interact. Intell. Syst., vol. 2, no. 3, p. 18, 2012.

[14] P. Burnap et al., "Tweeting the terror: Modelling the social media reaction to the Woolwich terrorist attack," Social Netw. Anal. Mining, vol. 4, no. 1, p. 206, 2014.

[15] N. Alsaedi, P. Burnap, and O. Rana, "Can predict a riot? Disruptive event detection using Twitter," ACM Trans. Internet Technol., vol. 17, no. 2, p. 18, 2017 .

[16] P. Burnap and M. L. Williams, "Us and them: Identifying cyber hate on Twitter across multiple protected characteristics," EPJ Data Sci., vol. 5, no. 1, p. 11, 2016.

[17] R. Delgado and J. Stefancic, "Hate speech in cyberspace," Wake Forest Law Rev., vol. 49, p. 319, Jan. 2014.

[18] K. Jaishankar, "Cyber hate: Antisocial networking in the Internet," Int. J. Cyber Criminol., vol. 2, no. 2, pp. 16–20, 2008.

[19] C. Schieb and M. Preuss, "Governing hate speech by means of counterspeech on Facebook," in Proc. 66th ICA Annu. Conf., Fukuoka, Japan, 2016, pp. 1–23.

[20] E. Chandrasekharan, M. Samory, A. Srinivasan, and E. Gilbert, "The bag of communities: Identifying abusive behavior online with preexisting Internet data," in Proc. ACM CHI, 2017, pp. 3175–3187.