I.J. Modern Education and Computer Science, **2018, 1, 46-56**
Published Online January 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2018.01.05

A Proposed Framework to Analyze Abusive Tweets on the Social Networks

Priya Gupta[1]
[1]Department of Computer Science, Maharaja Agrasen College, University of Delhi, Delhi, India
Email:[1] pgupta1902@gmail.com

Aditi Kamra[2], **Richa Thakral**[3], **Mayank Aggarwal**[4], **Sohail Bhatti**[5], **Vishal Jain**[6]
[2,3, 4, 5] Maharaja Agrasen College, University of Delhi, Delhi, India
[6]Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), Delhi, India
Email:[2]aditi.kamra9@gmail.com, [3]richathakral18@gmail.com, [4]mayank.agarwal.1994@gmail.com, [5]sbhatti1995@gmail.com, [6]drvishaljain83@gmail.com

Received: 22 September 2017; Accepted: 06 December 2017; Published: 08 January 2018

***Abstract*—**This paper takes Twitter as the framework and intended to propose an optimum approach for classification of Twitter data on the basis of the contextual and lexical aspect of tweets. It is a dire need to have optimum strategies for offensive content detection on social media because it is one of the most primary modes of communication, and any kind of offensive content transmitted through it may harness its benefits and give rise to various cyber-crimes such as cyber-bullying and even all content posted during the large even on twitter is not trustworthy. In this research work, various facets of assessing the credibility of user generated content on Twitter has been described, and a novel real-time system to assess the credibility of tweets has been proposed by assigning a score or rating to content on Twitter to indicate its trustworthiness. A comparative study of various classifying techniques in a manner to support scalability has been done and a new solution to the limitations present in already existing techniques has been explored.

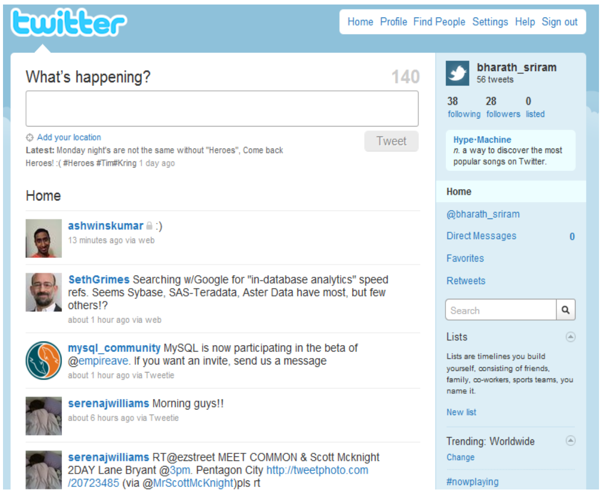
Index Terms—Twitter, Classifier, Detection, Semantic, Syntactic, Abusive, Data-Cleaning, Classification

I. Introduction

Online social media has evolved and gained much of popularity in the past few years. It serves as a medium which has a large reach and can be used by any person residing in any part of the world. It spans across barriers of country, religion, region, race and language. Currently, people are spending more and more time on social media to connect with others, to share a wide variety of information, and to pursue common interests (Churcharoenkrung et al, 2011) Seventy five percent of U.S. households (Diana, 2010) now use social networking sites; 83% of 18-29 year old (Ries, 2011) Americans are using social media, with 61% doing so every day. It has changed the way people access information or communicates.

One of the major online social micro blogging site is- Twitter. Twitter was found in March 2006 by Jack Dorsey. Since then it has grown exponentially and revolutionized the way people access information and news about current events (Lewis, 2004).Unlike traditional news media, online social media such as Twitter is a bidirectional media, in which common people also have a direct platform to share information and their opinions about the news events (Fig.1). It helps users to connect with other Twitter users around the globe. The messages exchanged via Twitter are referred to as micro-blogs because there is a 140 character limit imposed by Twitter for every tweet. This lets the users present any information with only a few words, optionally followed with a link to a more detailed source of information.

Therefore, Twitter messages, called as “tweets” are usually focused. In this regard, Twitter is very similar to SMS (Short Message Service) messages exchanged via mobile phones and other hand held devices. In fact, the 140-character limit on message length was initially set for compatibility with SMS messaging, and has brought to the web the kind of shorthand notation and slang commonly used in SMS messages. The 140 character limit has also spurred the usage of URL shortening services such as bit.ly, goo.gl, and tr.im, and content hosting services to accommodate multimedia content and text longer than 140 characters. Several other social networking sites like Facebook, Orkut introduced the concept of “Status” messages, some much before Twitter originated. But it was Twitter that went a step ahead and made these “statuses” be sharable between people through mobile phones since its creation. Users on Twitter, create their public / private profile and post messages (also referred as tweets or status) via the profile. Each post on Twitter is characterized by two main components: the tweet (content and associated metadata) and the user (source) who posted the tweet (Golbeck, 2005). People log onto Twitter to check for updates about events and also to share information about the event with others.



Latest: Monday night's are not the same without "Heroes", Come back Heroes! :(#Heroes #Tim#Kring 1 day ago

ashwinskumar :)
13 minutes ago via web

SethGrimes Searching w/Google for "in-database analytics" speed refs. Seems Sybase, SAS-Teradata, Aster Data have most, but few others!?
about 1 hour ago via web

mysql_community MySQL is now participating in the beta of @empireave. If you want an invite, send us a message
about 1 hour ago via Tweetie

serenajwilliams Morning guys!!
about 6 hours ago via Tweetie

serenajwilliams RT@ezstreet MEET COMMON & Scott Mcknight 2DAY Lane Bryant @3pm. Pentagon City http://tweetphoto.com/20723485 (via @MrScottMcKnight)pls rt

Hype·Machine
n. a way to discover the most popular songs on Twitter.

Lists are timelines you build yourself, consisting of friends, family, co-workers, sports teams, you name it.

Fig.1. Home Page of a Twitter user

In such a situation where Twitter has become an indispensable part of every common man's life it is necessary to filter out indecent or abusive content from the tweets that are being posted on Twitter as such tweets that can negatively affect the users especially adolescents. Furthermore, new press and studies have found that children and adolescents were engaged in producing online hate speech (Tynes et al., 2004), 3% of adolescents participated in cyber solicitation in 2008 (Finkelhor et al., 2008), and 13% of adolescents cyber-bullied others in 2010 (Hinduja & Patchin, 2008).To address this problem various software packages such as Appen and Security Suite have been developed but the precision of detection and filtration of offensive content they offer is very low (Churcharoenkrung et al, 2011). Also most of the existing software use words based approach which fail to detect a status as offensive if none of its terms are strongly offensive.

This paper aims to overcome the problems of the existing software's and architectures, and to compare the various techniques of data mining that can be employed to detect offensive content on Twitter (Frakes et al, 1992). First of all as there is no universal agreement as to what is "offensive". For the purpose of this paper, we employ Jay and Janschewitz (2008), definition of offensive language as vulgar, pornographic, and hateful language. Vulgar language refers to coarse and rude expressions, which includes explicit and offensive reference to sex or bodily functions, hateful language includes any communication outside the law that disparages a person or a group on the basis of his social reputation etc. (Golbeck, 2005). By comparing various techniques we would be able to identify which technique outperforms the other techniques under different constraints. In this paper we also tend to focus on both contextual as well as lexical aspect of the tweets in order to detect the offensive ones.

A. *Current Problems*

There are various challenges associated with real time content filtering in Twitter. Some of them are described below:

- **Volume of Content**: Most of the popular online social websites have users of the order of hundreds of millions. A huge amount of content is generated every second, minute and hour of the day (Boykin & Roychowdhury 2005).Any algorithms or solutions built to analyze the data should be highly scalable.
- **Diversity of content**: Twitter is one of the major and most used micro blogging site and contains views, opinions and information from every field make the data very diverse and highly unstructured. Language constraints may also be seen.
- **Real-time Analysis**: Impact of malicious activities in online social media, such as spread of spam, phishing or rumors, causes vast amount of damage within hours of being introduced in the media. Hence, solutions and algorithms built need to be able to solve and detect such content in real-time (Golbeck, 2005). Post-analysis may be able to capture concerned criminals, but would not be able to contain the damage.
- **Data Accessibility**: Due to privacy and anti-scraping policies of most OSM, data collection for research is a big challenge. Only public data can be extracted which often represents a very small percentage of the actual content, for e.g. Twitter and Face book. Though, APIs are provided by most of the social media websites, but rate limits are imposed on data collection via them, for e.g. a maximum of 350 requests per hour on Twitter Rest API.
- **Short Length of text**: Tweets posted on Twitter are generally short and do not contain many details. In order to analyze such tweets a short text classifier is needed to be implemented.
- **Spelling Mistakes**: People tend to do spelling mistake very often while posting tweets on Twitter which makes it difficult to carry out lexical and syntactic analysis of such words. Python libraries like enchant can be used but they do not provide full reliability for spelling correction.
- **Slang words**: People use various slang words and abbreviations in their tweets. Standardizing these slang words and abbreviations is a difficult task as there is no proper and reliable source of slang dictionary containing all slang words.
- **Contextual analysis:** Unlike lexical analysis there are no predefined approaches and rules to carry out the contextual analysis which pose a major challenge to diversify the tweets on the basis of context.

B. Problem Statement

This paper aims at building a model which presents comparison of various techniques to detect offensive tweets from Twitter on the basis of the content of the text of the tweet. People especially adolescent's benefit from use of social media by interacting with and learning from others, they are also at the risk of being exposed to large amounts of offensive online contents Twitter serves as a rich source of information. Unlike other information sources, Twitter is up-to-date and reflects the current news and events happening around the world. It has overpowered all the other major social networking sites.

But unfortunately in today's time social networking sites have become a major source of offensive content. ScanSafe monthly "Global Threat Report" found that up to 80% of blogs contained offensive contents and 74% included porn in the format of image, video, or offensive languages. In addition, cyber-bullying occurs via offensive messages posted on social media.

Hence, to combat this problem it is the need of the hour to have precise and accurate detection and filtration system to eliminate undesired or offensive tweets from Twitter. As there are number of techniques available for content filtering and it is important to analyze the technique having greatest precision and accuracy. One more important reason which makes comparison of each of the detection techniques necessary is that performance of each of these techniques (Naive Bayes', SVM etc.) are subject to constraints under which they are employed. With proper implementation of these techniques and on the social networking sites like Twitter, it would make these sites a more reliable and relevant sources of information and communication. This would further help in reducing the crime rates of cyber-crimes such as cyber bullying by a considerable number. In a nutshell, with sophisticated and accurate detection techniques Twitter can be used for the vision for which it was made - communication and entertainment.

C. Overview Of Proposed Solution And Benefits

Keeping the problem statement in mind as explained above, the proposed solution is as follows:

1. **Dataset**: This is preliminary requirement for the research. So, data is extracted using Twitter's tweepy streaming API. The tweets are extracted of a particular trending topic for detailed analysis. Thereafter, from the data extracted-the text and id of the tweets are extracted.
2. **Data Cleansing**: The collected dataset is processed to clean in order to reduce the analysis time and the overhead. All the data is converted to lowercase and stop words, slang words etc. are removed. The URLS and mentions in text are also removed as URLS and mentions are not required for analysis.
3. **Bag of Words**: Two special types of bags are created namely Offensive and Less Offensive. Offensive bag of words contain highly abusive words as per the Google's list of abusive words. The less offensive bag of words contain less offensive words which when seen individually might not be termed as offensive but when used with other words can be offensive.
4. **Contextual Feature Generation:** Two types of lists are generated which are used as training data for contextual analysis. One is a list of tweets extracted using the tweepy API. The tweets extracted contain the hashtags which are anti to the topic on which our testing dataset is based. The second one is of list of offensive adjectives from the most popular tweets which are anti to the topic of testing dataset. Popularity of the tweets is determined by the followers of the tweet. A tweet with more than 2000 followers is identified as popular.
5. **Feature Extraction**: Using the offensive bag of words, from the dataset- lexically offensive tweets and remaining tweets are extracted. Lexically Offensive tweets will be the one containing highly offensive words. Remaining tweets will be the one containing no offensive word but less offensive words.
6. **Lexical Analysis:** Further, from the remaining dataset the tweets are classified as lexically clean and lexically less offensive using the less offensive bag of words. Lexically clean tweets may be offensive on contextual basis.
7. **Machine Learning**: Naive Bayes` theorem is used with two to different types of clustering techniques for generating the input for Naive Bayes. A hybrid of LDA and Naive Bayes is also made.
8. **Contextual Classifier**: The contextual classifier compares two approaches of detecting contextually offensive content - cosine similarity and adjective based approach. The approaches use the contextual features generated.

There will be several benefits of using the proposed solution. Few of them are listed below:

1. Due to extensive cleaning and categorization of data, the processing time has decreased several folds.
2. The collected dataset is highly unstructured, hence contains a lot of slang words and spelling errors. To combat this problem the proposed solution standardizes all the slang words used and corrects the spelling using Edit Distance Algorithm.
3. The proposed solution offers a 3-tier architecture and emphasizes on both contextual and lexical aspects of the text of tweet.
4. Apache Spark is used to implement the various machine learning techniques for lexical analysis of data. As it is based on resilient dataset, it offers high processing speed and extensive support for data analytics.

5. The solution compares various classifiers approaches for lexical analysis. New Rule Based Naive Bayes and LDA Based Naive Bayes approach are used. Also no hybrid model of 2 Machine Learning techniques is used therefore, provides a new direction for thinking about the existing techniques.
6. For contextual analysis, a new adjective based approach has been proposed which offers higher recall and precision than the existing algorithms. As no previous work has been done on contextual analysis therefore, proposed approach throws new light on this aspect of text.

D. Existing Tools For Abusive Content Detection

Currently there are two types of tools predominantly present for this purpose which are as follows:

1. **Content Analysis Software**: Most commonly used content analysis software is Appen data stream profiling tools that capture online communication from different channels such as keystrokes, in-browser text editors etc. and generates alerts on pre-defined anomaly profiles.
2. **Parental Control Software:** tools such as Internet Security Suite, K9 Web Protection and on guard online can record children's online activity. It uses a pattern matching approach which determine the content offensiveness by detecting appearance of the predefined patterns such as words and phrases.

The main drawback of using these tools is word ambiguity problems. Both types of tools depend on pattern matching which generates high false positives.

E. Current Techniques For Offensive Content Detection

The current techniques for offensive content detection has been explained in the following table (Table 1).

Table 1. Taxonomy of previous studies based on three characteristics

CATEGORY	**DESCRIPTION**
Preprocessing methods	
Syntactic parsers	Natural language parser, Part-of-Speech
Semantic feature types	
Lexical feature	Bag of words-gram
Sentiment feature	Include pronoun , subjective terms
Contextual feature	Similarity feature , contextual post feature
User screening feature	Credibility of user on basis of his/her activity feature
Classification Approach	
Rule based approach	Keyword / phrase matching, rule based decision table
Machine learning approach	Support vector machine, Linear Discriminant Analysis, Naive Bayes' classifier-NN classifier, Decision tree, Random Forest Tree, Principal component analysis etc.

II. Requirement Specifications, Analysis, Design And Modelling

The proposed system requires to classify the offensive content on twitter using the lexical as well as contextual aspect of tweets. For the above stated objective it is required to do extensive data cleaning so as to reduce the load on further components of the system. This requires standardizing the slang words as the tweets are highly unstructured. Various other data cleaning techniques also needs to be employed. In the feature generation phase it is required to use standardized lexical features so that they can be applied independently, irrespective of the content of the tweets. The contextual features generated required threshing under various constraints such as popularity of the feature.

The lexical analyzer requires extensive training of training model so as to give precise results. The context based classifier requires proper preprocessing and vectorisation of the text of the fields. For this it requires proper lemmatization, stemming and removal of stop words.

III. Context, Functional And Non-Functional Requirements

This section describes how requirements are organized across the document through the following categories of requirements:

- List of actors;
- Context requirements – i.e., generic "environment" requirements;
- Functional requirements;
- Non-functional requirements.

A. List of Actors:

Data curator, model trainer, efficiency evaluation, model predictor, Discrepancy evaluator, database etc.

B. Context Requirements

Table 2. Context Requirements

S.No.	**Requirement Name**	**Description**
1.	Data Import from Legacy Sources	Process of extraction of data from CA proprietary products requires a number of data importing capabilities.
2.	The system should be able to protect privacy of data.	Data fields that contain personally identifiable (personal and corporate) information should be obfuscated or encrypted with appropriate access control restrictions.
3.	The system should be able to consider different model types.	In order to model the behavior of certain parameters of the system, the system should be able to allow for different types of models. Different types of models may include for instance time series regressions, event-condition action rules, neural networks, etc.

The specific context or "environment" requirements generated are listed below (Table 2):

C. *Functional Requirements*

The specific functional requirements generated are listed below (Table 3):

Table 3. Functional Requirements

S.No.	Requirement Name	Description
1.	Data Extraction requirements	● Twitter API : REST service with tweepy ● Re : for preprocessing of data using string matching patterns ● Enchant : Spell checker ● Nltk library : for removing stop words
2..	Data Visualization requirements	● Comparison charts between classifiers. ● matplotlib: visualize the data analysis ● Python libraries: NUMPY. SCIPY
3.	Contextual analysis requirements	● Ipython notebook ● Pandas ● Apache Spark ● NLTK corpus ● Stemmers and lemmatize
4..	Lexical analysis requirements	● Ipython notebook ● NLTK Stop words ● Apache Spark ● PySpark ● Ml.Classification ● Gensim

D. *Non-Functional Requirements*

The non-functional requirements, especially targeting Big Data specific issues, are further subdivided in the following categories and has been explained in Table 4.

- Performance
- Integration
- Stability
- Maintainability
- Scalability

Table 4. Non-Functional Requirements

S. No.	Requirement Name	Description
Performance Requirements		
1.	Data import/export size	The system should be able to import/export tens of terabyte data volumes.
2.	No data loss in streams	The system should be able to store and process the streams received or generated without dropping any data.
3.	Latency of real time query	Latency of database queries should not negatively affect user experience and usability
4.	Interactive latency	Initial drawing of visualization should not take more than 2 seconds. Update of visualization should not take more than 250 milliseconds Click delay during interaction with visualization no more than 2 seconds.
Integration Requirements		
5.	Capacity to deal with heterogeneous data sources	The system should be ready to cope with heterogeneous data sources, extracted from different proprietary sources, and integrate them so that they can be used together in an integrated way during the analysis process.
Stability Requirements		
6.	Past data should be available	Data should be available in the system for long periods to perform further analysis in the future that are not defined at this time of the project
Maintainability Requirements		
7.	Update of models	We should be able to delete, modify or update the models created at any time
8.	Updates of data sources	We should be able to update the data sources used in our system.
Scalability Requirements		
9.	System should be able to scale in case we need to monitor many different metrics and models	We may need to monitor several models related to different data sources. The system should be prepared to scale up in case the number of monitored sources grows significantly
10.	Visualizations should be able to scale for the display of large data sets on multiple devices	We may need to visualize datasets of multiple sizes and different metrics from heterogeneous data sources. The visualization system must be capable of presenting usable visualization on diverse devices such a mobile devices and large displays.

IV. Overall Proposed Architecture

The three level architecture of the proposed solution consists of 10 major components. Their interrelationship has been describe in Fig.2. Each of them is briefly explained below

- **Data Acquisition**: By using the twitter REST API tweepy, the system communicates with twitter in order to extract tweets.
- **Data Cleaning Unit**: In this component of the architecture, the raw tweets extracted using the Tweepy REST API are pre-processed so that they could be further classified by the feature extraction unit.
- **Lexical Feature Generation Unit**: In this unit two types of bag of words are created. One contains the highly offensive words and the second one contains the words that when seen individually are not highly offensive but when used with other offensive words, nouns and proverbs, they tend to be highly offensive. This will help in classifying the tweets as lexically offensive and lexically clean.
- **Contextual Feature Generation Unit**: In this unit two types of lists are generated which are

used as training data for contextual analysis. One is a list of tweets extracted using the tweepy API. The tweets extracted contain the hash tags which are anti to the topic on which our testing dataset is based. The second one is of list of offensive adjectives from the most popular tweets which are anti to the topic of testing dataset

- **Feature Extraction Unit**: In this component the tweets that are retrieved from the data cleaning unit are lexically classified using the highly offensive bag of words created in lexical Feature Generation Unit.
- **Lexically Clean Unit**: This unit of architecture depicts or stores the lexically clean tweets i.e. the tweets that doesn't contain any of the offensive words but contains phrases that can be contextually offensive. This unit further sends the tweets to the context based classifier for further classification.
- **Lexically Offensive Unit:** This unit stores the lexically offensive tweets i.e. the tweets containing the offensive words. This unit further sends the tweets for further classification to the lexical and context based classifier.
- **Context Based Classifier**: When this unit coupled with the lexically clean unit, it further classifies the tweets as neutral and non-neutral tweets on the basis of the subjective content of the tweet by using natural language processing approaches.
- **Analytics Unit**: In this unit the various machine learning and natural language processing techniques used to build classifiers are compared on the basis of precision, recall and accuracy.

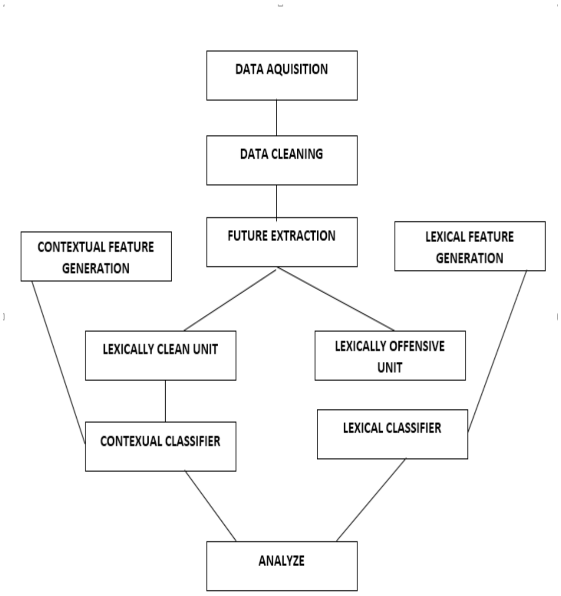


Fig.2. Proposed Architecture

V. Design Documentation

The design documentation of the proposed system has been explained in terms of Use Case Diagram (Fig.3.) and Control Flow Diagram (Fig. 4.)

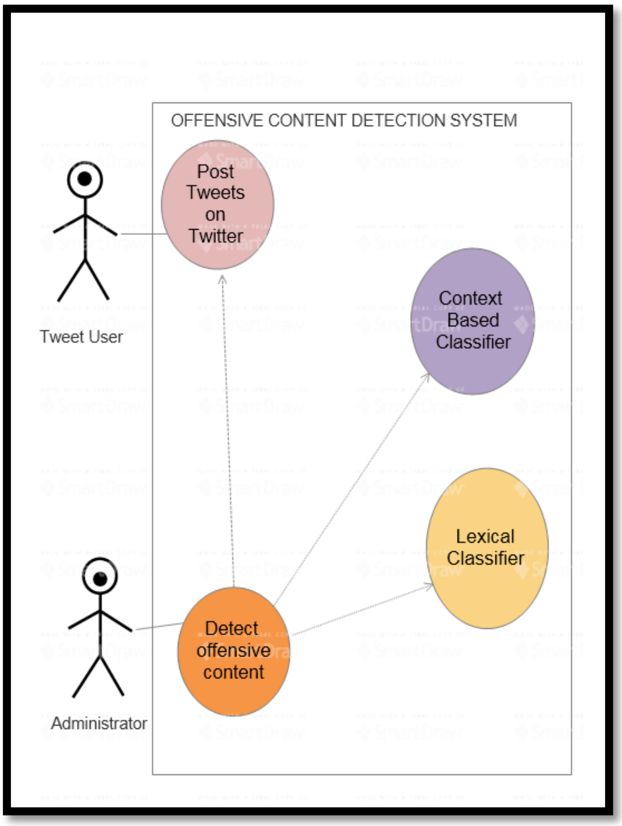


Fig.3. Use Case Diagram of Proposed Solution

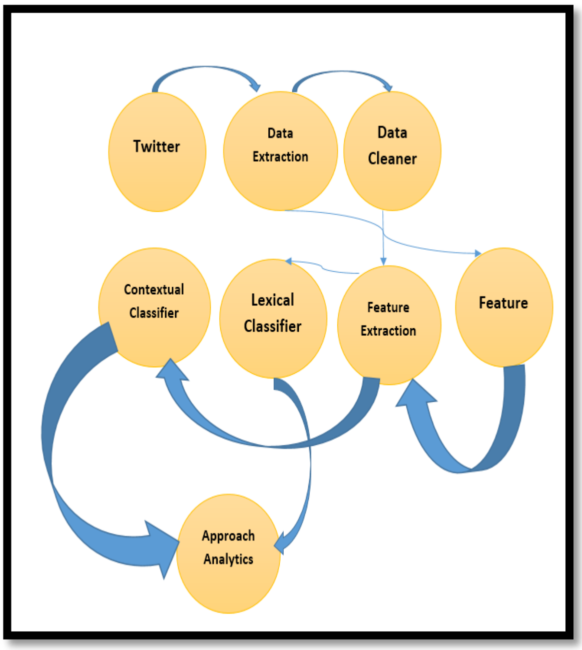


Fig.4.Control Flow Diagram of Proposed Solution

VI. Implementation

The step wise step algorithm of the proposed system has been explained further and their interrelationship is shown in Fig.5.

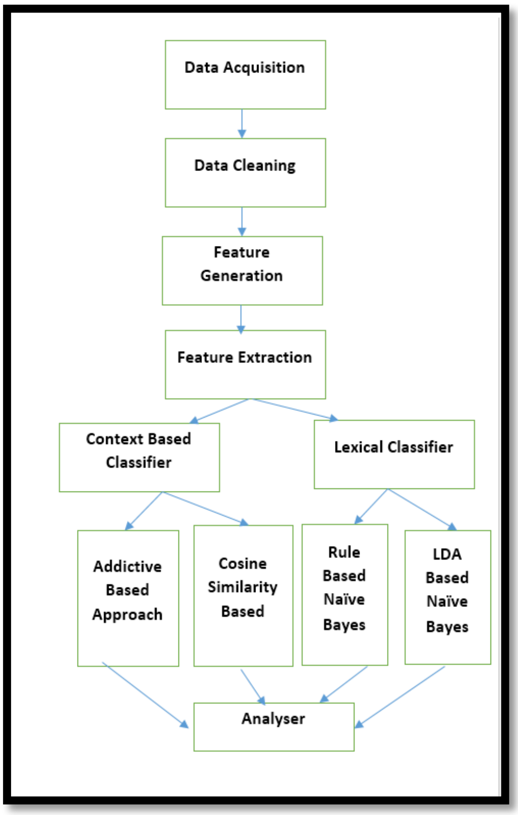


Fig.5. Step wise step algorithm of the Proposed Architecture

A. Data Acquisition

In data acquisition phase, Tweets are extracted from the twitter REST API - Tweepy. The extracted tweets are stored in a text file. To maintain uniformity dataset was collected of the currently trending topic. In this paper tweets so collected were about Mr. Donald Trump. The text file containing the tweets is converted into an apache pyspark resilient distributed dataset (RDD).The tweets so extracted are containing various parameters of information such as tweet_id, location, date of creation, text of the tweets etc. Hence to decrease the overhead of processing the tweets the text and id parameters are extracted from the tweets. As the size of the dataset is quite large the extracted texts of the tweets are then converted to the apache resilient dataset. This is done so as to increase the processing speed of the tweets.

B. Cleaning Of Raw Tweets

Natural language parsing techniques are used to pre-preprocess the raw tweets. Python's re and String class are used for this purpose (Fig.6.).

- **Lower Case:** The tweets are converted to lowercase letters to remove the ambiguity.
- **Standardizing Slang Words**: Slang words dictionary is created which contains various slang words and their standardized words. This dictionary is used to standardize all the slang words present in the text of the tweet.
- **Replacement of Urls and User Tags**: URL links and @user are replaced by the words url and at user.
- **Removal of Punctuation Marks**: punctuation marks and extra spaces are removed from the tweets.
- **Removal of Repeated Characters**: As most of the times tweet contain informal type of data and people therefore tend to use characters repeatedly. For example, awesomeeeeeeeee, lovelyyyyy. Such words are standardized by removing these repeated occurrence of same character.
- **Removal of Stop Words**: A dictionary containing common stop words such as a, the etc. is created. Then using this dictionary stop words are removed from the tweets.
- **Spelling Correction**: At the last stage of preprocessing the incorrect spellings are corrected using python's Enchant library and edit distance algorithm.

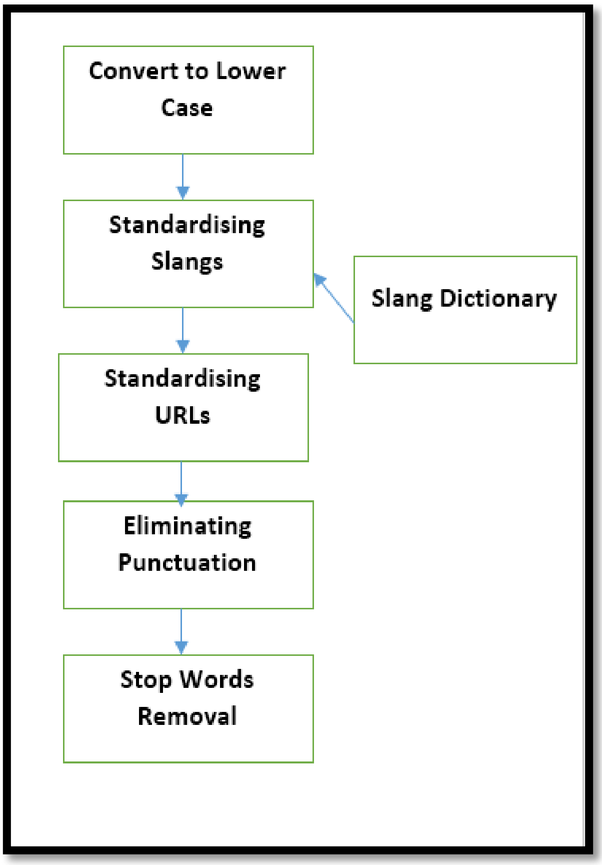


Fig.6. Data Cleaning Process

C. *Feature Generation*

For lexical analysis two types of bag of words (BOW) are created (Table 5). First one contains the highly offensive words(BOW-1) and the other contains the less offensive words(BOW-2) which when used with other offensive words or nouns or proverbs then can be termed as highly offensive. For example, phrase such as “this is a stupid thing” cannot be termed as offensive but a phrase like “you are so damn stupid” where stupid is being used with a second person can be termed as offensive.

For this purpose, dictionaries available on the web such as wikidoc, eureka etc. will be parsed using python’s beautiful HTML soup.

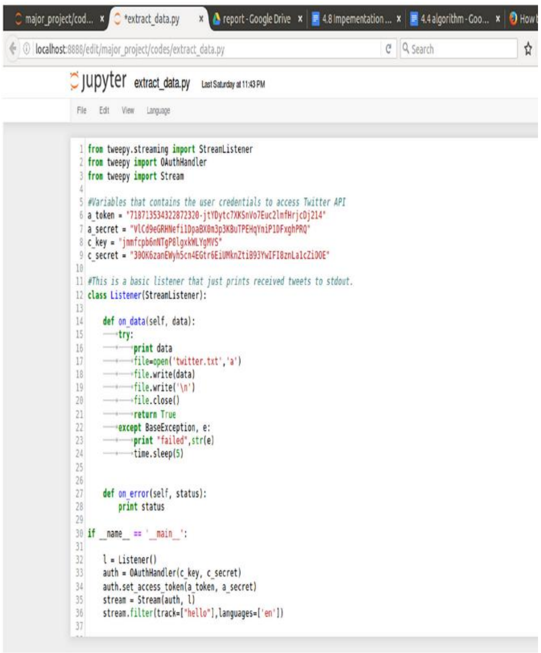
Further these dictionaries will be stored as apache RDD and will be cached in the memory so as to save the space

Table 5. Types of bag of words used

TYPES OF BOW	EXAMPLES
TYPE – 1	Fuck, fucking etc
TYPE – 2	Stupid, silly etc

For contextual analysis two types of lists are generated which are used as training data for contextual analysis. One is a list of tweets extracted using the tweepy API. The tweets extracted contain the hashtags which are anti to the topic on which our testing dataset is based. The second one is of list of offensive adjectives from the most popular tweets which are anti to the topic of testing dataset. The tweets anti to the topic was extracted though the most used anti hashtags for Donald Trump such as “#notMyPresident”, “#Antitrump”. Popularity of the tweets is determined by the followers of the tweet. A tweet with more than 2000 followers is identified as popular.

D. *Feature Extraction*

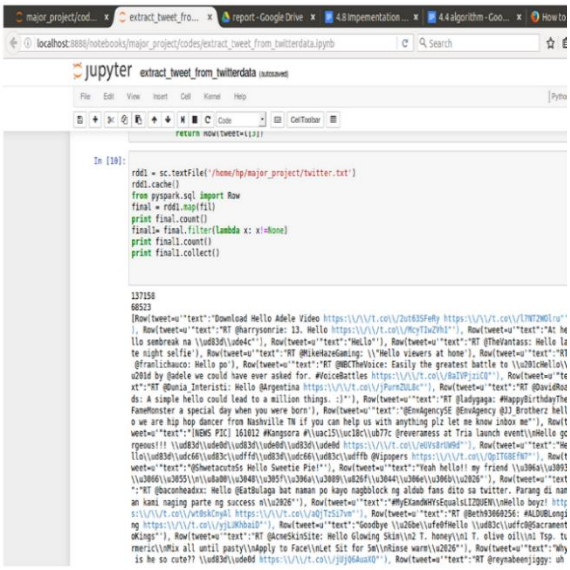


```
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream

#Variables that contains the user credentials to access Twitter API
a_token = "718713534322872320-jtYDytc7XKSnVo7Euc2lmfHrjcDj214"
a_secret = "VlCd9eGRHNefi1DpaBX0m3p3KBuTPEHqYniP1DFxghPRQ"
c_key = "jmnfcpb6nNTgP8lgxkWLYgMVS"
c_secret = "390K6zanEWyh5cn4EGtr6EiUMknZtiB93YwIFI8znLa1cZiD0E"

#This is a basic listener that just prints received tweets to stdout.
class Listener(StreamListener):

    def on_data(self, data):
        try:
            print data
            file=open('twitter.txt','a')
            file.write(data)
            file.write('\n')
            file.close()
            return True
        except BaseException, e:
            print "failed",str(e)
            time.sleep(5)


    def on_error(self, status):
        print status

if __name__ == '__main__':

    l = Listener()
    auth = OAuthHandler(c_key, c_secret)
    auth.set_access_token(a_token, a_secret)
    stream = Stream(auth, l)
    stream.filter(track=["hello"],languages=['en'])

```

Fig.7. Extracting Data

In the feature extraction phase 1, the preprocessed tweets are then classified as highly offensive. For this the BOW-1 created in feature generation phase is used. If the tweets contain any of the words that are present in BOW-1, then they are classified as offensive.

In the feature extraction phase 2, the remaining preprocessed tweets are then classified as lexically clean and lexically offensive. For this the BOW-2 created in feature generation phase is used. If the tweets contain any of the words that are present in BOW-2, then they are classified as lexically offensive (Fig.7, Fig.8. Fig.9.).


major_project/cod...
extract_tweet_fro...
report - Google Drive
4.8 Impementation ...
4.4 algorithm - Goo...
How to
localhost:8888/notebooks/major_project/codes/extract_tweet_from_twitterdata.ipynb
Search
jupyter extract_tweet_from_twitterdata (autosaved)
File Edit View Insert Cell Kernel Help
Code
CellToolbar
Pytho

```
        return Row(tweet=l[3])

In [10]:
rdd1 = sc.textFile('/home/hp/major_project/twitter.txt')
rdd1.cache()
from pyspark.sql import Row
final = rdd1.map(fil)
print final.count()
final1= final.filter(lambda x: x!=None)
print final1.count()
print final1.collect()

137158
68523
[Row(tweet=u'"text":"Download Hello Adele Video https:\\/\\/t.co\\/2ut63SFeRy https:\\/\\/t.co\\/l7NT2WOlru"'
), Row(tweet=u'"text":"RT @harrysonrie: 13. Hello https:\\/\\/t.co\\/McyTIwZVh1"'), Row(tweet=u'"text":"At he
llo sembreak na \\ud83d\\ude4c"'), Row(tweet=u'"text":"HeLlo"'), Row(tweet=u'"text":"RT @TheVantass: Hello la
te night selfie'), Row(tweet=u'"text":"RT @MikeHazeGaming: \\"Hello viewers at home'), Row(tweet=u'"text":"RT
 @franlichauco: Hello po'), Row(tweet=u'"text":"RT @NBCTheVoice: Easily the greatest battle to \\u201cHello\\
u201d by @adele we could have ever asked for. #VoiceBattles https:\\/\\/t.co\\/8aIVPjziCQ"'), Row(tweet=u'"te
xt":"RT @Dunia_Interisti: Hello @Argentina https:\\/\\/t.co\\/jPurmZUL8c"'), Row(tweet=u'"text":"RT @DavidRoa
ds: A simple hello could lead to a million things. :)"'), Row(tweet=u'"text":"RT @ladygaga: #HappyBirthdayThe
FameMonster a special day when you were born'), Row(tweet=u'"text":"@EnvAgencySE @EnvAgency @JJ_Brotherz hell
o we are hip hop dancer from Nashville TN if you can help us with anything plz let me know inbox me"'), Row(t
weet=u'"text":"[NEWS PIC] 161012 #Kangsora #\\uac15\\uc18c\\ub77c @reveramess at Tria launch event\\nHello go
rgeous!!! \\ud83d\\ude0d\\ud83d\\ude0d\\ud83d\\ude0d https:\\/\\/t.co\\/eUVs8rUW9d"'), Row(tweet=u'"text":"He
llo\\ud83d\\udc66\\ud83c\\udffd\\ud83d\\udc66\\ud83c\\udffb @Vipopers https:\\/\\/t.co\\/OpITG8EfN7"'), Row(t
weet=u'"text":"@ShwetacuteSs Hello Sweetie Pie!"'), Row(tweet=u'"text":"Yeah hello!! my friend \\u306a\\u3093
\\u3066\\u3055\\n\\u8a00\\u3048\\u305f\\u306a\\u3089\\u826f\\u3044\\u306e\\u306b\\u2026"'), Row(tweet=u'"text
":"RT @baconheadxx: Hello @EatBulaga bat naman po kayo nagbblock ng aldub fans dito sa twitter. Parang di nam
an kami naging parte ng success m\\u2026"'), Row(tweet=u'"text":"#MyEXandWHYsEqualsLIZQUEN\\nHello boyz! http
s:\\/\\/t.co\\/wt0skCnyAl https:\\/\\/t.co\\/aQjTzSi7vm"'), Row(tweet=u'"text":"RT @Beth93060256: #ALDUBLongi
ng https:\\/\\/t.co\\/yjLUKhbaiD"'), Row(tweet=u'"text":"Goodbye \\u26be\\ufe0fHello \\ud83c\\udfc0@Sacrament
oKings"'), Row(tweet=u'"text":"RT @AcneSkinSite: Hello Glowing Skin\\n2 T. honey\\n1 T. olive oil\\n1 Tsp. tu
rmeric\\nMix all until pasty\\nApply to Face\\nLet Sit for 5m\\nRinse warm\\u2026"'), Row(tweet=u'"text":"Why
 is he so cute?? \\ud83d\\ude0d https:\\/\\/t.co\\/jUjQ6AuaXQ"'), Row(tweet=u'"text":"RT @reynabeenjiggy: uh
```

Fig.8. Extracting texts from Tweets

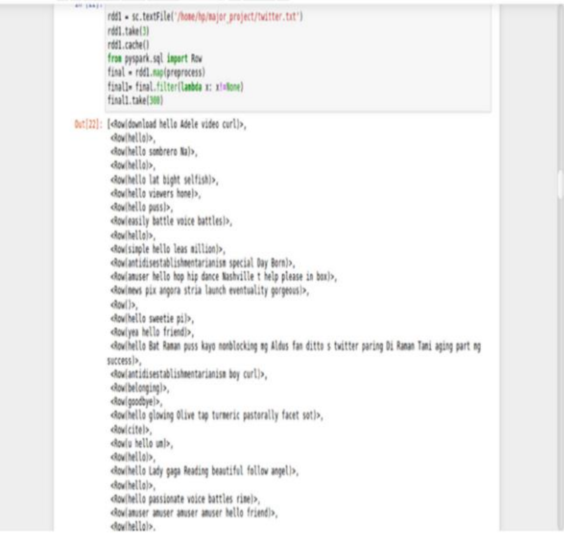
E. *Classification Of Lexically Clean Tweets*

The contextual classifier uses two natural language processing approaches of detecting contextually offensive content - cosine similarity and adjective based approach. The approaches use the contextual features generated.

The tweets being tested are first converted into vector form using term frequency. The cosine similarity approach uses the tweet list feature generated in feature generation phase as training data. It compares the similarity of the tweet being processed with the training data. If the cosine similarity between the tweet being tested and any of the tweet in the training dataset is greater than 60% then that tweet is classified as offensive. Cosine similarity between 2 vectors v1 and v2 are computed in (1).

$$\theta = v1.v2 \div (|v1||v2|) \tag{1}$$

A new natural language processing based approach is developed for contextual analysis. For the proposed method, the adjective feature generated in the feature generation phase is used to classify the tweets. The tweets containing any of these adjectives are classified as offensive. The rest remaining tweets are classified as neutral tweets (Fig.10.).



```
rdd1 = sc.textFile('/home/hp/major_project/twitter.txt')
rdd1.take(3)
rdd1.cache()
from pyspark.sql import Row
final = rdd1.map(preprocess)
final1= final.filter(lambda x: x!=None)
final1.take(300)
```

```
Out[22]: [<Row(download hello Adele video curl)>,
 <Row(hello)>,
 <Row(hello sombrero Na)>,
 <Row(hello)>,
 <Row(hello lat bight selfish)>,
 <Row(hello viewers hone)>,
 <Row(hello puss)>,
 <Row(easily battle voice battles)>,
 <Row(hello)>,
 <Row(simple hello leas million)>,
 <Row(antidisestablishmentarianism special Day Born)>,
 <Row(amuser hello hop hip dance Nashville t help please in box)>,
 <Row(news pix angora stria launch eventuality gorgeous)>,
 <Row()>,
 <Row(hello sweetie pi)>,
 <Row(yea hello friend)>,
 <Row(hello Bat Raman puss kayo nonblocking mg Aldus fan ditto s twitter paring Di Raman Tami aging part mg
success)>,
 <Row(antidisestablishmentarianism boy curl)>,
 <Row(belonging)>,
 <Row(goodbye)>,
 <Row(hello glowing Olive tap turmeric pastorally facet sot)>,
 <Row(cite)>,
 <Row(u hello um)>,
 <Row(hello)>,
 <Row(hello Lady gaga Reading beautiful follow angel)>,
 <Row(hello)>,
 <Row(hello passionate voice battles rime)>,
 <Row(amuser amuser amuser amuser hello friend)>,
 <Row(hello)>.
```

Fig.9. Tweets after Preprocessing

VII. Conclusion

From the above comparative study of context based classifiers it was found that the newly devised adjective based approach outperforms the existing cosine similarity based approach in terms of recall and precision.

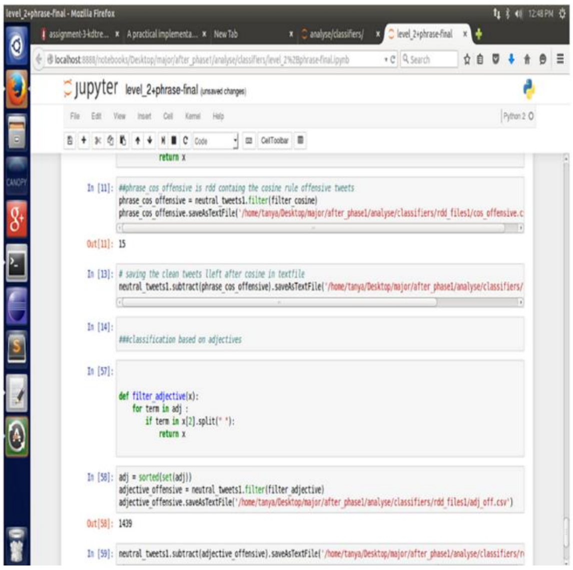
It can be concluded that the adjective based approach is better for context based analysis of tweets when the offensive content volume is low in the test dataset.

The main reason for this is as the adjective based approach only takes in consideration the most important part of speech of text for classification i.e. adjectives, it will always perform optimally irrespective of the volume of the offensive content in the test data.

The newly proposed approach poses some problems such as dependency on the topic of the test data to accumulate the training data. Hence, it needs further threshing by adding more constraints so as to further increase the precision and accuracy.

The cosine similarity based approach has a upper hand in terms of accuracy of classification. But this approach offers very low precision and recall with nearly unacceptable values. Hence it can be concluded that cosine similarity based approach is not optimal for context based classification when the volume of offensive content is low.

With any other methodology for vectorisation of tweets such as tf-idf weighting it may be possible to increase the precision and recall of cosine similarity based approach.



level_2+phrase-final - Mozilla Firefox
12:48 PM
assignment-3-kdtre...
A practical implementa...
New Tab
analyse/classifiers/
level_2+phrase-final
localhost:8888/notebooks/Desktop/major/after_phase1/analyse/classifiers/level_2%2Bphrase-final.ipynb
Search
jupyter
level_2+phrase-final (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
CellToolbar
CANOPY

```
        return x

In [11]: ##phrase_cos_offensive is rdd containg the cosine rule offensive tweets
         phrase_cos_offensive = neutral_tweets1.filter(filter_cosine)
         phrase_cos_offensive.saveAsTextFile('/home/tanya/Desktop/major/after_phase1/analyse/classifiers/rdd_files1/cos_offensive.c

Out[11]: 15

In [13]: # saving the clean tweets lleft after cosine in textfile
         neutral_tweets1.subtract(phrase_cos_offensive).saveAsTextFile('/home/tanya/Desktop/major/after_phase1/analyse/classifiers/

In [14]:
         ###classification based on adjectives

In [57]:

         def filter_adjective(x):
             for term in adj :
                 if term in x[2].split(" "):
                     return x

In [58]: adj = sorted(set(adj))
         adjective_offensive = neutral_tweets1.filter(filter_adjective)
         adjective_offensive.saveAsTextFile('/home/tanya/Desktop/major/after_phase1/analyse/classifiers/rdd_files1/adj_off.csv')

Out[58]: 1439

In [59]: neutral_tweets1.subtract(adjective_offensive).saveAsTextFile('/home/tanya/Desktop/major/after_phase1/analyse/classifiers/r
```

Fig. 10. Context based Classifier

VIII. Limitations of Proposed Solution

As the proposed solution proposes new approaches for contextual as well as lexical analysis, it still poses few limitations. The contextual analysis' dataset depends upon the topic of which the tweets are extracted. The training dataset consists of the tweets and adjectives anti to the topic of which the tweets are collected. Though the overall processing time of the proposed solutions has decreased several folds, but it is still considerably large.

Though few rules were decided but there may be several other rules that may be possible and may provide better results.

References

[1] Bonchi, F., Castillo, C., Gionis, A., & Jaimes, A. (2011). Social network analysis and mining for business applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 22.

[2] Boykin, P. O., & Roychowdhury, V. P. (2005). Leveraging social networks to fight spam. *Computer*, IEEE Computer Magazine, *38*(4), 61-68.

[3] Chelmis, C., & Prasanna, V. K. (2011, October). Social networking analysis: A state of the art and the effect of semantics. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on* (pp. 531-536). IEEE.

[4] Churcharoenkrung, N., Kim, Y. S., & Kang, B. H. (2005, April). Dynamic Web content filtering based on user's knowledge. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on* (Vol. 1, pp. 184-188). IEEE.

[5] Frakes, W., Baeza-Yates, R. (eds.) (1992),Information Retrieval: Data Structures & Algorithms, *Prentice-Hall*

[6] Gavrilis, D., Tsoulos, I., & Dermatas, E. (2006). Neural recognition and genetic features selection for robust detection of e-mail spam. *Advances in Artificial Intelligence*, 498-501.

[7] Golbeck, J. A. (2005). *Computing and applying trust in web-based social networks* (Doctoral dissertation).

[8] Kim, Y. H., Hahn, S. Y., & Zhang, B. T. (2000, July). Text filtering by boosting naive Bayes classifiers. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 168-175). ACM.

[9] Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, *5*(Apr), 361-397.

[10] Schütze, H. (2008, June). Introduction to information retrieval. In *Proceedings of the international communication of association for computing machinery conference*.

[11] Venkata S. Lakshmi, K. Hema,(2014) Filtering Information for Short Text Using OSN *International Journal of Advanced Research in Computer Science & Technology (IJARCST*), 2(2)

[12] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, *24*(5), 513-523.

[13] Bobicev, V., & Sokolova, M. (2008, July). An Effective and Robust Method for Short Text Classification. In *AAAI* (pp. 1444-1445).

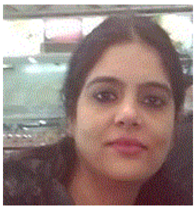
[14] Yerazunis, W. S. (2004, January). The spam-filtering accuracy plateau at 99.9% accuracy and how to get past it. In *Proceedings of the 2004 MIT Spam Conference*.

[15] Zavuschak, I & Burov, Y.(2017), "The Context of Operations as the basis for the Construction of Ontologies of Employment Processes", *International Journal of Modern Education and Computer Science(IJMECS),* 9(11), 13-24, DOI: 10.5815/ijmecs.2017.11.02

[16] Narinder K. Seera, Vishal Jain,"Perspective of Database Services for Managing Large-Scale Data on the Cloud: A Comparative Study", *IJMECS*, vol.7, no.6, pp.50-58, 2015.DOI: 10.5815/ijmecs.2015.06.08

[17] Asad Mehmood, Abdul S. Palli, M.N.A. Khan,"A Study of Sentiment and Trend Analysis Techniques for Social Media Content", *IJMECS*, vol.6, no.12, pp.47-54, 2014.DOI: 10.5815/ijmecs.2014.12.07

Authors' Profiles



Dr. Priya Gupta is working as an Assistant Professor in the Department of Computer Science at Maharaja Agrasen College, University of Delhi. Her Doctoral Degree is from BIT (Mesra), Ranchi. She has more than 13 years of teaching and 5 years of Industry Experience. Her research Interest lies in the area of Machine Learning, Theory of Computation, Compiler Design, Data Mining, Artificial Intelligence etc. She has authored book titled "CRM System and Cross Selling in Indian Banking Industry", "Innovation in Payment Systems – An Approach Towards Cashless Mandis", and Banking the Unbanked - A Step Towards Financial Inclusion in Indian Mandis. She has also edited book titled "Innovative Minds – Technocrats with Vision" (Unfolding the dimensions of Compiler Design and Microprocessor) - Volume I and "Innovative Minds – Technocrats with Vision" (Unfolding the dimension of Artificial Intelligence and Information Security) - Volume – II.

Aditi Kamra has completed her B.Tech in Computer Science from MaharajaAgrasen College, University of Delhi, India in 2017. Presently she is student ambassador at M.A.P.

Richa Thakral has completed her B.Tech in Computer Science from Maharaja Agrasen College, University of Delhi, India in 2017. She received numerous awards in presenting research papers. She received meritorious award from RajyaSabha , Parliament House, New Delhi. She is currently working with an MNC.

Mayank Aggarwal has completed her B.Tech in Computer Science from Maharaja Agrasen College, University of Delhi, India in 2017. Presently he is working with Hitesh Industries.

Sohail Bhatti has completed her B.Tech in Computer Science from Maharaja Agrasen College, University of Delhi, India in 2017.

Dr. Vishal Jain is currently working as Associate Professor with Bharati Vidyapeeth's Institute of Computer Applications and Management (BVICAM), New Delhi Affiliated to GGSIPU and Accredited by AICTE, since July, 2017 to till date. He has joined BVICAM, New Delhi in year 2010 and worked as Assistant Professor from August, 2010 to July, 2017. Before joined BVICAM, New Delhi, he has worked four years in Guru Presmsukh Memorial College of Engineering, Affiliated to GGSIPU and Accredited by AICTE, from July 2004 to July, 2008. Dr. Vishal Jain has completed Ph.D (Computer Science and Engineering) from Lingaya's University, Faridabad, Haryana, M.Tech (Computer Science and Engineering) from University School of Information Technology (USIT), Guru Gobind Singh Indraprastha University, MBA (HR) from Shobhit University, Meerut, MCA from Sikkim Manipal University, Sikkim. In additional qualification he has obtained DOEACC ‘A’ Level and DOEACC ‘O’ Level, Post Graduate Diploma in Computer Software Training from A.M Informatics, Advance Diploma in Computer Software Training from ET&T, Delhi, Diploma in Business Management from All India Institute of Management Studies, Chennai, Diploma in Programming from Oxford Computer Education, Delhi, Microsoft Certified Professional Cleared Two Modules 070-210, 070-215 (MCP) and Cisco Certified Network Administrator (CCNA). He has received Young Active Member award for the year 2012 – 13 from Computer Society of India. His research area includes Web Technology, Semantic Web and Information Retrieval. He is Life member of CSI and ISTE.

How to cite this paper: Priya Gupta, Aditi Kamra, Richa Thakral, Mayank Aggarwal, Sohail Bhatti, Vishal Jain, "A Proposed Framework to Analyze Abusive Tweets on the Social Networks", International Journal of Modern Education and Computer Science(IJMECS), Vol.10, No.1, pp. 46-56, 2018.DOI: 10.5815/ijmecs.2018.01.05