

CUSTOMER REQUIREMENT SPECIFICATION


 <p>PES UNIVERSITY</p>		Document Ref.:		
		Version No.:		
		Date	:	20 February 2019
Project Name:		Detection of Events and Emerging Themes from Social Media Streams		
Project Code:		PW19COP01		
Status:		Draft / Current / Superseded		
Document Type:		Controlled / Uncontrolled		
Detection of Events and Emerging Themes from Social Media Streams				
Detecting events of national importance, disease outbreaks, and other emergencies and displaying the same on a user's Twitter feed.				
Prepared By:		Reviewed By:		
Name	Date	Name	Date	
Vaishnavi Rao B USN: 01FB15ECS334	20/02/2019			
Varsha R USN: 01FB15ECS337	20/02/2019	Approved By:		
		Name	Date	
Distribution List				
Project Representative(s)		Guide Representative(s)		
1. Vaishnavi Rao B 2. Varsha R		3. Prof. C. O. Prakash		

TABLE OF CONTENTS

Definitions, Acronyms and Abbreviations	3
References	3
Change History	4
1.0 Introduction	5
1.1 Scope	5
2.0 Product Perspective	5
2.1 User Characteristics	5
2.2 General Constraints, Assumptions and Dependencies	5
2.3 Risks	5
3.0 System Architecture	6
4.0 Requirements List	6
4.1 Module / Scenario 1	6
4.2 Module / Scenario 2	6
4.3 Module / Scenario n	6
5.0 External Interface Requirements	7
5.1 Hardware Requirements	7
5.2 Software Requirements	7
5.3 Communication Interfaces	7
6.0 User Interfaces	7
7.0 Performance Requirements	7
8.0 Special Characteristics	8
9.0 Help	8

10.0	Other Requirements	8
10.1	Site Adaptation Requirements	8
10.2	Safety Requirements	8
11.0	Packaging	8
12.0	Traceability Matrix	9

References

- [1] Catching the Long-Tail: Extracting Local News Events from Twitter; Puneet Agarwal, Rajgopal Vaithiyanathan, Saurabh Sharma and Gautam Shroff

- [2] Topic Extraction from News Archive Using TF*PDF Algorithm; Khoo Khyou Bun Mitsuru Ishizuka

- [3] A Survey of Techniques for Event Detection in Twitter; Farzindar Atefeh and Wael Khreich

- [4] Sentiment-Based Event Detection in Twitter; Georgios Paltoglou

- [5] Emerging Event Detection in Social Networks with Location Sensitivity; Unankard Sayan, Xue Li, and Mohamed A Sharaf

- [6] Predicting Crowd Behaviour with Big Public Data; N. Kallus

- [7] Event Detection in Social Media Data; Wenwen Dou, Xiaoyu Wang, William Ribarsky, Michelle Zhou

- [8] Tweet Analysis for Real-Time Event Detection and Earthquake Reporting System Development Ekta, Paahuni Khandelwal, Priya Bundela, Richa Dewan

Change History

This section describes the details of changes that have resulted in the current CRS document.

#	Date	Document Version No.	Change Description	Reason For change
1.				
2.				
3.				

1.0 Introduction

The micro-blogging platform, Twitter, has opened people up to an entirely new world of information-gathering and news-sharing, including but not limited to detailing everyday events and activities of the users, the “Tweeple”. Twitter, then, becomes an exhaustive source of data for analysis as well as the first place people are increasingly turning to for daily alerts or updates, in the face of news organisations and other traditional forms of media people looked to earlier. A major challenge facing event detection using social media streams is to extract useful real-world events from the mundane and polluted text (abbreviated words, spelling and grammatical errors and text written in mixed languages). [3] Event detection requires the automatic answering of what, when, where, and by whom. After reporting on the most recent efforts in the area, it is clear that no method addressed all of these questions. [5] Investigating how to model the social streams together with other data sources, like news streams to better detect and represent events was another complication faced earlier. [6] A considerable effort is still required to achieve efficient, scalable and reliable systems for event detection, summarization, tracking and association. [7]

1.1 Scope

The project will depend on the efficient functioning of Twitter minus any crashes on any given day to be able to conduct real time analysis smoothly in the long run. A proper, working internet connection is required that keeps the end user connected with their surroundings in order to access Twitter easily. Additional limitations include false positives generated by the model and disinformation provided by people on Twitter. Twitter users are assumed to be sensors, who make observations, which are their actual tweets to detect a target event. The tweets are given a time values and a location as well. The detection of these events and other emerging topics is restricted to the Indian subcontinent.

2.0 Product Perspective

- This project is dependent on Twitter and its users. It is not functionable without Twitter’s huge fanbase and customers.
- Twitter in general has the following features :-
 - o A home page where there are a bunch of tweets in textual format displayed to a Twitter user with a valid Twitter account.
 - o This home page has tweets from people the user follows.
 - o There is another section of each user’s account with general trending tweets decided by Twitter’s algorithm based on the most used hashtags and most liked/shared tweets.
 - o There is also a user profile that displays activities of the user, the user’s tweets and retweets.
 - o Identifying the principal external interfaces of this software product.
- Python and a simple laptop with 8GB RAM is used for development
- Python, Bootstrap and a simple laptop with 8GB RAM is used for deployment

2.1 User Characteristics

The haphazard nature of tweets can make a timeline quite messy. Twitter also happens to be one of the most important sources of news for people where they hear or read about a particular event first. In the face of this, we propose a solution where the user's timeline also reflects trending topics based on national emergency and other filters among other trending topics. This not only pushes events of national importance to the top of the user's timeline, it also filters out certain other trending topics that are given importance by media organisations in which the user might not be all that interested.

2.2 General Constraints, Assumptions and Dependencies

Dependencies:

- The project will depend on the efficient functioning of Twitter minus any crashes on any given day to be able to conduct real time analysis smoothly in the long run. A proper, working internet connection that keeps the end user connected with their surroundings in order to access Twitter easily is another network-related issue the successful functioning of the model is dependent on.

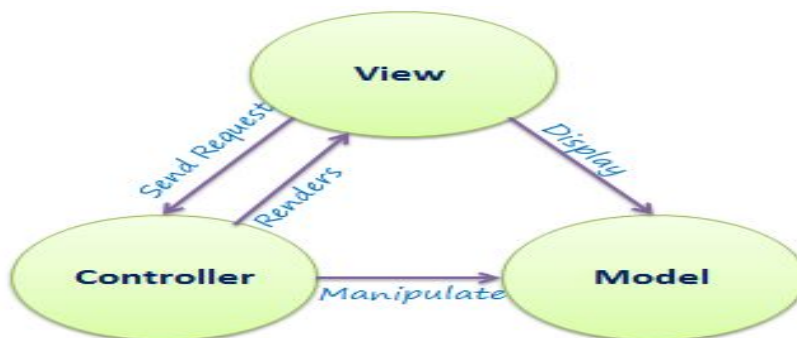
Assumptions:

- Each Twitter user is assumed to be a sensor, that detects a target event and makes an observation. This observation is the actual tweet.
- The tweets are given time values as well as a location (latitudinal and longitudinal coordinates).

2.3 Risks

- Misinformation given by people is the biggest threat that the project faces.
- Additionally, false positives generated by the model can pose a problem that will cause confusion among the users.

3.0 System Architecture



- Preprocessing the tweets using simple nltk tools
- Tweets of that particular day will be displayed. This will be fetched directly from twitter.com from current 'yyyy:mm:dd 00:00:00' timestamp till the current time 'yyyy:mm:dd hh:mm:ss'

- on clicking on the trending topic, most relevant tweets from the residing country (India) with respect to natural disasters, health hazards (epidemic and breakouts), terror attacks and severe weather forecasts will be displayed. These will be the only 4 categories.
- If no relevant trending tweets were found that belong to any of these categories, the tweets will be displayed as usual.
- We will plan on adding a default sorting of the tweets based on the likes/shares of the tweets.

Architecture:

We plan to follow the MVC system architecture

1. Model: we will have a business unit with basically our fixed twitter training data. The twitter live data fetched from the website will be our testing data, both stored in our database.
2. View: we do require a frontend view to display to the users. Although there isn't much interaction with the interface, it is required to hide the implementation from the end user and for simple display of resultant tweets
3. Controller: since we have number of algorithms that are being tested on the tweets, a voting mechanism is adopted at the backend and the best algorithm is chosen to display the outputs to the user interface.

4.0 Requirements List

4.1 Module / Scenario 1

Reqmt #	Requirement
CRS – 1	Generates the trending topics of the day after being fed the tweets of the day. Tweets of that particular day will be displayed. This will be fetched directly from twitter.com from current 'yyyy:mm:dd 00:00:00' timestamp till the current time 'yyyy:mm:dd hh:mm:ss'

4.2 Module / Scenario 2

Reqmt #	Requirement
CRS – 1	Figure out the best algorithm to use by testing it on various machine learning and data science models and comparing accuracies (eg:- naive bayes, RNN, Random Forest, SVM, etc)

4.3 Module / Scenario n

Reqmt #	Requirement
CRS – 1	Adhere to customization of tweets displayed on the feed based on the trending topics. On clicking on the trending topic, most relevant tweets from the residing country (India) with respect to natural disasters, health hazards (epidemic and breakouts), terror attacks and severe weather forecasts will be displayed. These will be the only 4 categories.

5.0 External Interface Requirements

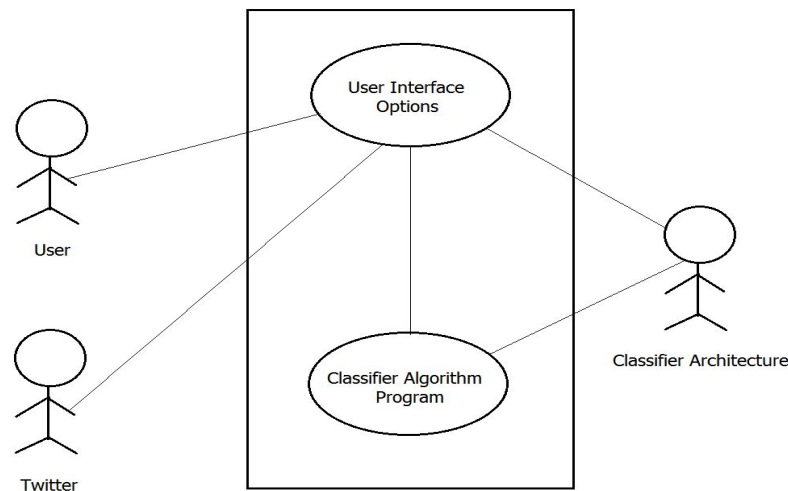
5.1 Hardware Requirements

Any laptop with a decent internet connection will be able to use our software.

5.2 Software Requirements

- Python 3.7
- Anaconda navigator
- Pandas
- Scikit-learn
- Keras

6.0 User Interfaces



- We will be providing a simple user interface with the most recent tweets filtered in accordance with the option chosen by the user
- Twitter data from Kaggle is being used for training and live tweets from twitter.com are fetched for testing. We will be using Sentiment140 dataset. It contains 1,600,000 tweets extracted using the twitter api . The tweets have been annotated (0 = negative, 2 = neutral, 4 = positive) and they can be used to detect sentiment . It contains the following 6 fields:
 - target: the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive)
 - ids: The id of the tweet (2087)
 - date: the date of the tweet (Sat May 16 23:58:44 UTC 2009)
 - flag: The query (lyx). If there is no query, then this value is NO_QUERY.
 - user: the user that tweeted (robotickilldozr)
 - text: the text of the tweet (Lyx is cool)

We will only be using the text column of the dataset

7.0 Performance Requirements

- 1 terminal for the user to look at comparison of different algorithms and their performances and 1 for display of tweets via the UI
- 1 user can use the UI. No login is required.
- The entire day's tweets need to be processed. Could be above 500MB to 4GB
- System only caters to tweets generated in the Indian subcontinent
- 1.6 million tweets will be used for training in the ratio of 80:20 for train test split. After validation, live tweets from twitter.com will be fetched for users.

8.0 Special Characteristics

- A user's login credentials with their username and password as will be provided in the UI will be the security check that will be required to prevent any unauthorized access. Additionally, the backend database will be accessible only to the managers on the backend.
- The dataset will be stored in the servers as mentioned, and the logs of the tweets will be maintained there itself.

9.0 Help

A simple about page will be added for users to understand what our website does and explains the idea behind this project.

10.0 Other Requirements

10.1 Site Adaptation Requirements

- The website that acts as the frontend to display the tweets to a user will be updated in a timely manner to collect all the tweets relevant to a given category.
- The website will also adhere to the latest standards and versions followed by *twitter.com*.

10.2 Safety Requirements

The final website, in keeping with Twitter India's safety standards, will adhere to policies recommended by CSR India. Furthermore, users will be prevented from giving away sensitive information such as their bank details and other national identity numbers. Repeated posting and duplicate tweets, as well as multiple tweets with unrelated links will be marked as spams.

11.0 Packaging

Given that the product aims to collate tweets from different users, the media and method of sharing is going to be packaged as a social media platform itself, and will maintain the look and feel of one as well.

12.0 Traceability Matrix

REQUIREMENTS TRACEABILITY MATRIX				
Project Name:	Detection of Events and Emerging Themes from Social Media Streams			
Reviewer/Approver	Vaishnavi Rao/Varsha R			
Traceability #	Requirement ID	Categorisation	Status	Test Case ID
a	Tweet_001			1
b	Tweet_002			2
c	Tweet_003			3
b	Tweet_004			4

The above Matrix defines the status of the tweets along with their categorisation, with the status reflective of whether they have been successfully categorised, or whether there is an error, or if any changes are to be made. For reference, only four have been included so far.

URS Reference Section No. and Name	CRS Reference Section No. and Name

- The CRS is the basis for changes in specifications or requirements of the design in the project. It should be reflected in the Change History section.
- The CRS should state requirements and constraints clearly and concisely. Design details should not be included in the CRS.