**Call For Code 2019 – DRAS\_Power Deadline – 06/07/2019**

**Project Title**

Categorization & mining tweets for disaster management

**Project Description**

Recent trend has shown that tweeter is an emerging source for monitoring disaster events. The objective of this project is to use supervised machine learning technique to automatically mining and classifying the tweet messages into different phases of a disaster that are generated during a course of natural disaster. An initial effort has spent for manually labelling 1,000 tweet messages into different themes to build a classification model for the predefined categories. During the initial annotation, process findings are referenced from the official government procedures [National Disaster Management Plan (NDMP)] in order to determine categories. Thereafter, an automatic program is developed to categorize the subsequent tweets into different themes. The five classes or categories considered in this work are as preparedness, response, impact, recovery and other. Once the classification is done, next level objective is to gain a quicker understanding of the categories of messages within each disaster phase, impacted location, volunteer relief organization, whether that message is in complain mode etc.

**Current Problem -**

Twitter can be used as an effective real time public communication platform between nongovernmental organizations (NGOs), public, government agencies, ad-hoc volunteer groups to share, exchange and collaborate critical information fast during a course of disaster. Social media data streams can be used to mine actionable data for emergency response and relief operation. However, manual examination of millions of unstructured social media posts is a challenge for disaster management in time of crisis.

**Solution Description –**

Machine Learning could be used to classify tweets into various stages of a disaster. Categorized and mined tweets are useful for emergency managers to identify the transition between phases of disaster management, that save their time to organize information, relief efforts and streamline priorities and activities during the course of a disaster. Entity extraction is further used to clue rescue workers in to the hardest hit areas and people in the most need.

**Included Components -**

* IBM cloud Services Used:

1. Natural Language Understanding-2t
2. Watson Studio-ra
3. natural-language-classifier-am

* IBM cloud storage Used:

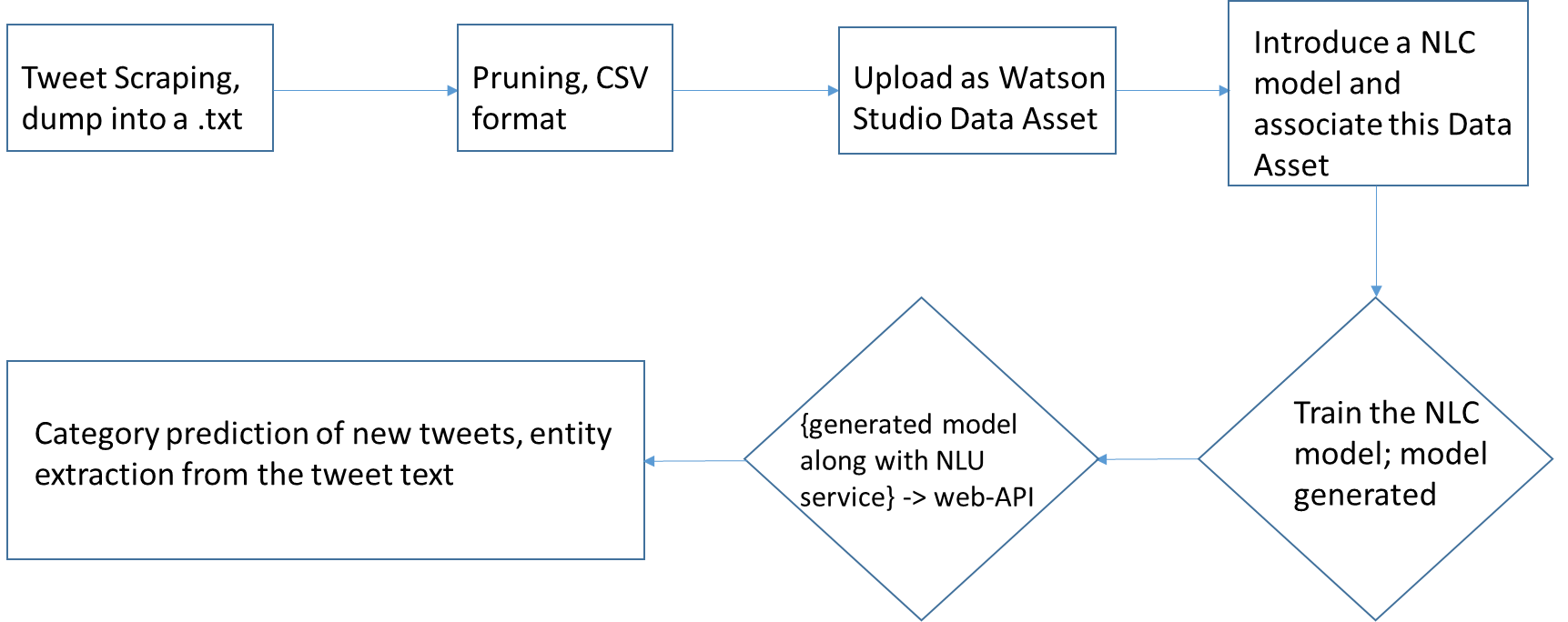
1. cloud-object-storage-yw

* IBM Cloud Foundry App

**Featured Technologies –**

* Python (version: 3.6.5)
* Flask (version: 1.0.2)
* Pyquery (version: 1.4.0)
* watson\_developer\_cloud (version: 2.10.1)
* xlwt (version: 1.3.0)

**Architecture Diagram -**



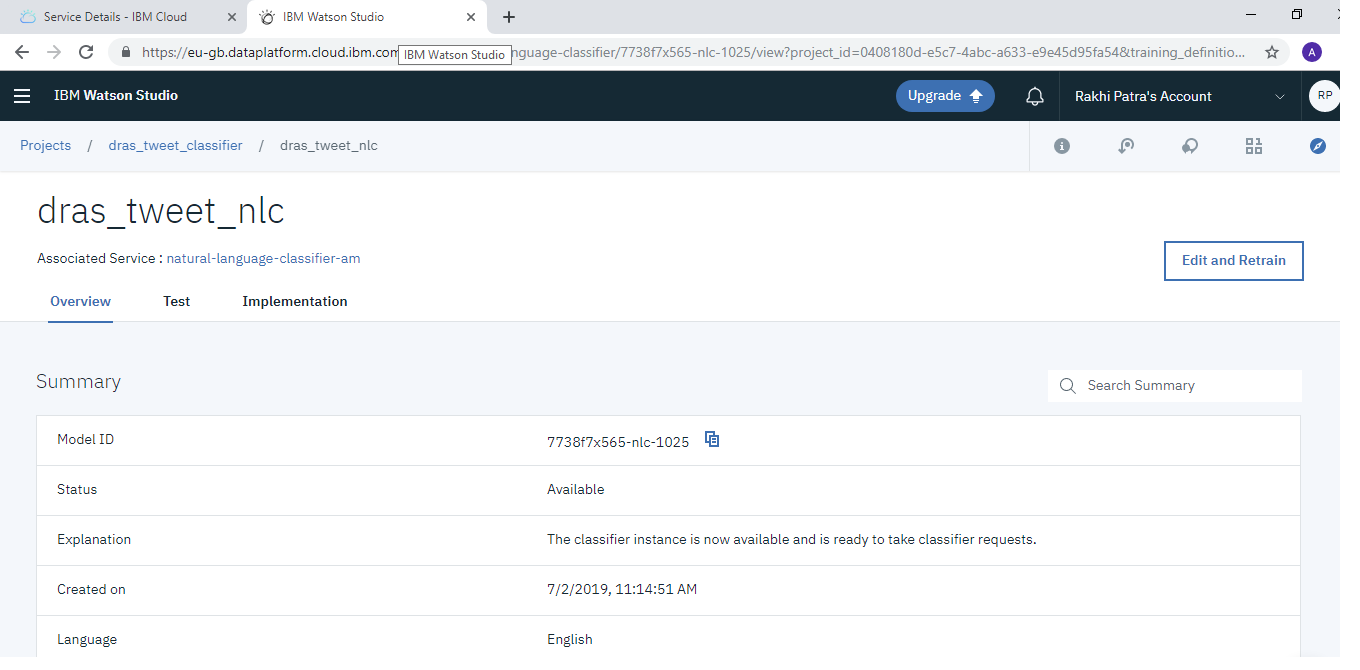
**Steps -**

We adhere the following steps.

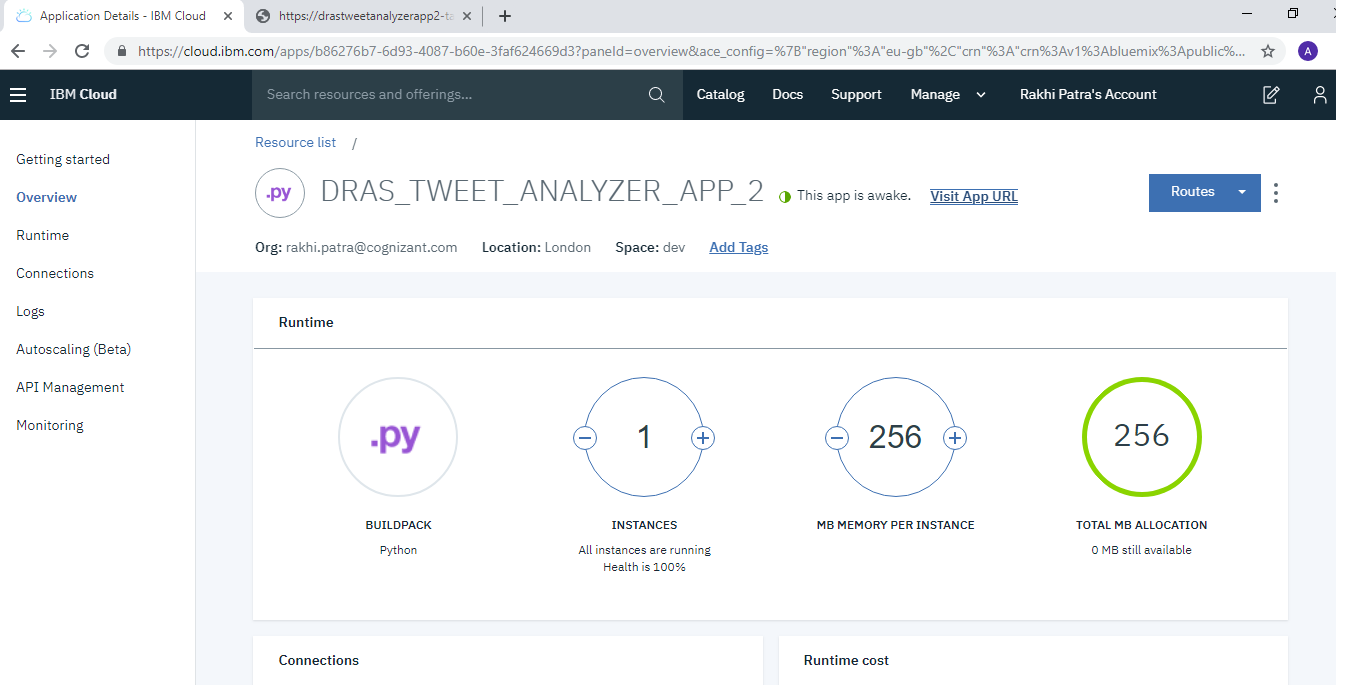
1. At first, tweets are being scraped with a keyword (#hashtag). Our twitter scraping code is independent of using anyone’s twitter username and password. Tweets are scraped for a particular duration and are dumped into a text file.
2. Next pruning of the tweet dump file is done manually, to treat this file as our training file to be used in natural language classifier. As the tweets fall under casual sentence without proper grammatical norms, so the tweet dump file has been pruned to adjust with inverted commas. Secondly, all the tweets are kept with a csv format, as provided by the example of IBM clould natural language classifier.

*Note: IBM Cloud natural language classifier service can’t deal with tweets directly and that’s why manual pruning is needed, mainly to adjust the inverted comma. This is a bottleneck, we have found.*

1. We maintained five categories for all the tweets what are to be used for training of NLC. The categories came at the last for each tweet in the training file. So the csv format becomes – {tweet-text, category}.
2. After successful uploading the training file (.txt) at IBM Cload Watson Studio as a Data Asset, we did the association of it with the NLC model.
3. Then, NLC model training has been initiated and it took some time to complete the training. After, successful training, the model id will be generated with available status. The screenshot is shared below.



1. The generated NLC trained model has been used to predict the category of the new tweets.
2. Next, IBM Cloud natural language understanding has been used to further dig out various entities from a tweet. Various text analytics features (category, entities &sentiment) have been extracted as a data mining process and as the result of it, sentiment, location, organization/company, topic have found.
3. The whole process of tweet categorization (by using NLC model) and text analytics features extraction (by using NLU service) have been wrapped to create an API. The code has been written in python.
4. Finally, the created API has been exposed to IBM Cloud as IBM Cloud Foundry App (Our generated Cloud Foundry App is “DRAS\_TWEET\_ANALYZER\_APP\_2”), so that it can be used as a web-API. This is done by using “flask” framework using python coding. The screen shot is shared below.

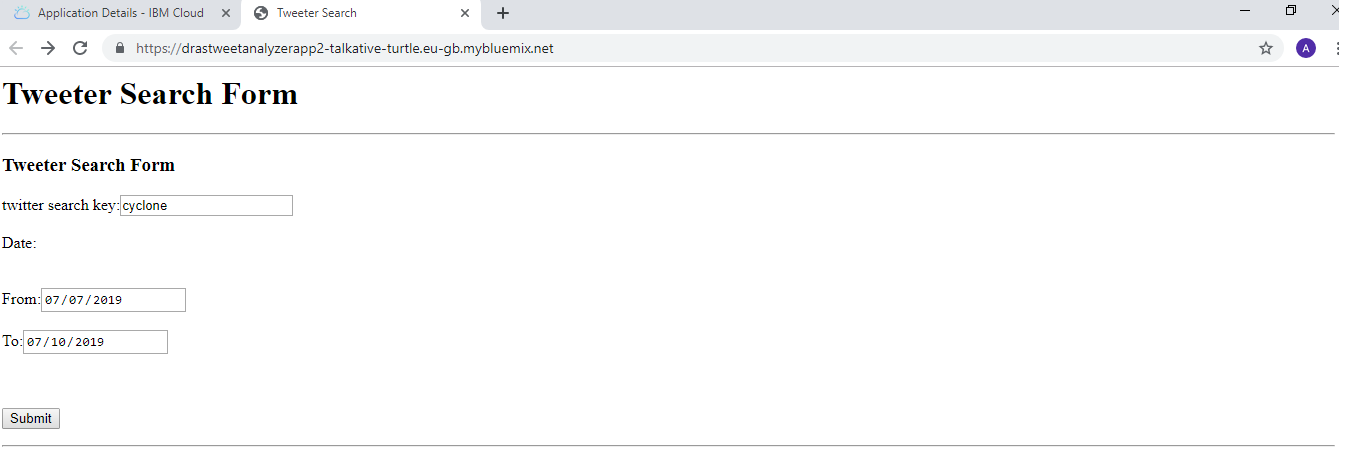


**Results –**

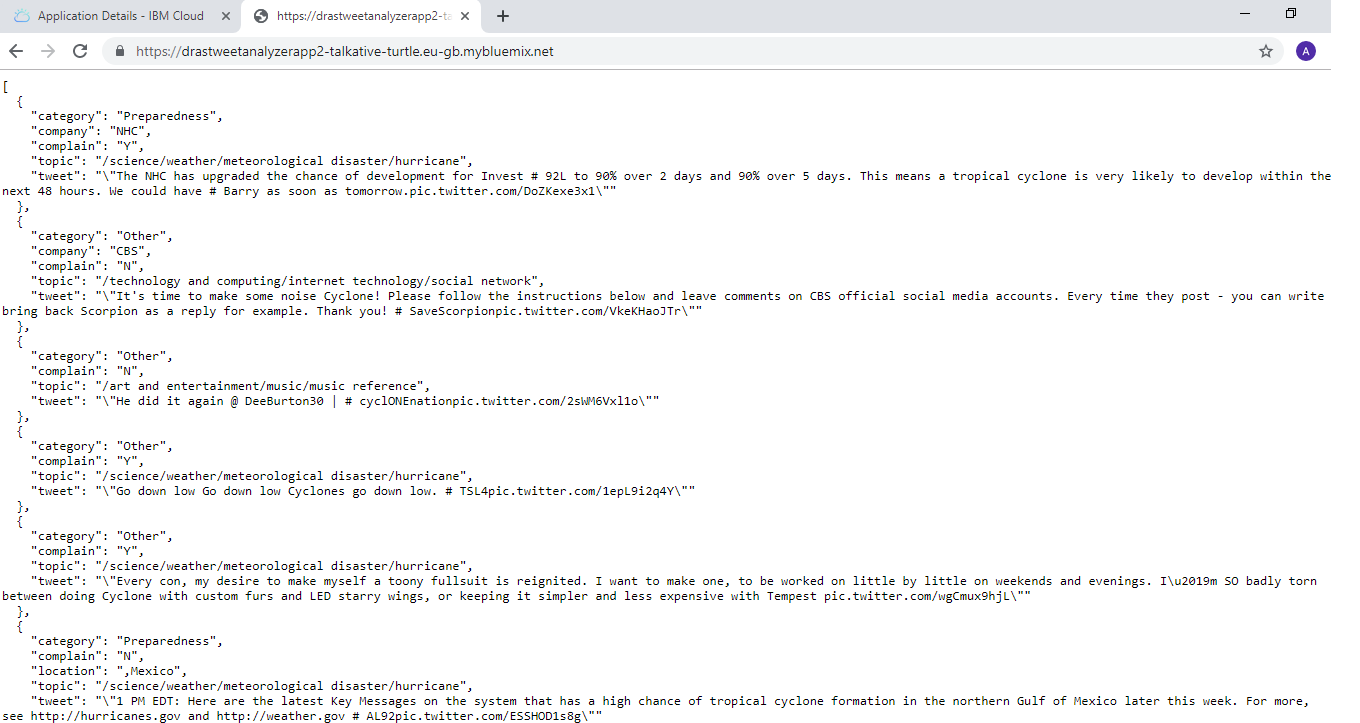
1. Our generated web-API is mentioned below:

<https://drastweetanalyzerapp2-talkative-turtle.eu-gb.mybluemix.net/>

Once the above mention URL is hit, UI is open for tweet search:



The tweet search result is shared below.



Notes:

* + In the above example, a restriction has been made only for **ten tweets**, as the IBM Cloud “Lite” plan is being used and the number of transactions for each service is limited.

1. Once the tweet search is done, the analysis report can also be downloaded in terms of an excel file by appending “download” after the above-mentioned URL; the download URL is mentioned below.

<https://drastweetanalyzerapp2-talkative-turtle.eu-gb.mybluemix.net/download>

As the result of this, “tweet\_analysis.xls” file will be downloaded. Sample output is shared below.

**

Future Improvement scope

* The idea is to run the tweet classifier program as a continuous process (scheduler) during a course of disaster. Future plan is to provide a UI so that user can configure the schedule time and the duration.
* Scrape additional information from tweets like originator of tweets, user type (Individuals, Celebrities, Journalists, News organisations, Government or NGOs), date & time of tweets.
* Currently we have used standalone IBM NLU service in order to identify category of tweet message. In future, we will build our own machine-learning model for customized category.