1) The project runs in python.

2) It require following python dependencies:

a) TWEEPY python library: required for fetching the data according to some filters

sudo apt-get install easy\_install

easy\_install tweepy   
 pip install tweepy

b) SUBPROCESS python library: required for updating the information obtained from computations on a web-server (where we have our data).

c) NLTK: For performing Natural Language Processing

3) Other dependencies:

SQLite3: Required to creating/reading the database of words for computation purposes

For developing visualization we require many tools and frameworks like rooby on rails, postgre sql etc., but they are not required for a demo, because we provide a direct web-link through which Visualization can be seen.

4) Different Code Files present/required in the project:

a) “createDB.py” : Creates the databse with name wordBase.db

b) “synonyms.py” : It was used to get the synonyms of the words which we obtained from Stanford’s, Harvard’s words and their sentiment scores. Just to enhance the database.

c) “wordBase.db” : Database of words and their sentiment scores

d) “getTweets\_DVA.py” : Main code, which executes every thing

5) Code flow:

getTweets\_DVA.py :

(a) obtains the Tweets, location and other details from Twitter using Tweepy Python library.

(b) Mines the data, gets useful information and tokenize the tweets

(c) Performs NLP and other computation, gets sentiment score

(d) uses sentiment score, location and tweet together to be uploaded on a server (table name: “exp\_geo”):

<http://clokar.cartodb.com/tables/exp_geo/public>

(e) We use postgresql to upload/update the data to the table: “exp\_geo”.

(f) Real-time Visualization is generated in the back-end, tweets are plotted on top of world map with sentiment score.

6) To run the code do : python getTweets\_DVA.py

While you run the code, it will open many Mozilla Firefox tabs, and run queries to upload the data retrieved from Twitter and after performing computation. There might be some exceptions while uploading and handling the data, but they are taken care of and that will not break the code.

7) Visualization can be seen at:

<https://clokar.cartodb.com/viz/1afc2242-cb42-11e3-b1a8-0e230854a1cb/public_map?title=true&description=true&search=false&shareable=true&cartodb_logo=true&layer_selector=false&legends=true&scrollwheel=true&fullscreen=true&sublayer_options=1&sql=&zoom=0&center_lat=52.482780222078205&center_lon=260.52978515625>

Various parameters of the Torque visualization can be configured through CartoCSS (CSS tab provided on the right ) or through the menu driven user interface (at the visualization wizard)

8) Entire code is dynamic and real-time, it will update the data which would be reflected on top of map.