# Prep work

a.Dataset from a set of sample Twitter data contained in a compressed (.zip) folder here:

http://s3.amazonaws.com/hw-sandbox/tutorial13/SentimentFiles.zip

b. Upload via Hue to HDFS

c. Download Json serial deserialer for Hive table upload as data is in JSON format

# Terminal Console

hadoop fs -copyToLocal /data/serde/json-serde-1.3-jar-with-dependencies.jar

In Hadoop file system have the json data on tweets, Open Hive and set the below parm to prevent hive checking for keywords so that fields/tables can have keyword 'name; etc.

set hive.support.sql11.reserved.keywords=false;

add jar json-serde-1.3-jar-with-dependencies.jar;

list jars;

# HUE Editor

create database 2017grc0\_twitter\_sentiment;

use 2017grc0\_twitter\_sentiment;

CREATE EXTERNAL TABLE tweets\_raw (id BIGINT,

created\_at STRING,

source STRING,

favorited BOOLEAN,

retweet\_count BIGINT,

retweeted\_status STRUCT<text:STRING,users:STRUCT <screen\_name:STRING,name:STRING>>,

entities STRUCT<urls:ARRAY<STRUCT <expanded\_url:STRING>>,

user\_mentions:ARRAY<STRUCT<screen\_name:STRING,name:STRING>>,

hashtags:ARRAY<STRUCT<text:STRING>>>,

text STRING,

user STRUCT<screen\_name:STRING,name:STRING,friends\_count:INT,followers\_count:INT,statuses\_count:INT,verified:BOOLEAN,utc\_offset:STRING,time\_zone:STRING>,in\_reply\_to\_screen\_name STRING, year BIGINT, month BIGINT, day BIGINT, hour BIGINT) ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe' WITH SERDEPROPERTIES ("ignore.malformed.json" = "true")

LOCATION '/user/YOUR-USERNAME/SentimentFiles/SentimentFiles/upload/data/tweets\_raw' ;

In Hue ,select Query editor and add the following

ADD JAR hdfs:////data/serde/json-serde-1.3-jar-with-dependencies.jar

CREATE table for Dictionary for the keywords and the sentiments (positive , negative)

CREATE EXTERNAL TABLE dictionary (

type string,

length int,

word string,

pos string,

stemmed string,

polarity string

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

LOCATION '/user/YOUR-USERNAME/SentimentFiles/SentimentFiles/upload/data/dictionary';

CREATE table for time zone with time-zone , country and comments

CREATE EXTERNAL TABLE time\_zone\_map (

time\_zone string,

country string,

notes string

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'

STORED AS TEXTFILE

LOCATION '/user/n\_ganapathy20037603/SentimentFiles/SentimentFiles/upload/data/time\_zone\_map';

CREATE VIEW tweets\_simple AS

SELECT

id,

cast ( from\_unixtime( unix\_timestamp(concat( '2013 ', substring(created\_at,5,15)), 'yyyy MMM dd hh:mm:ss')) as timestamp) ts,

text,

user.time\_zone

FROM tweets\_raw

;

Joining time\_zone\_map with tweets\_simple

CREATE VIEW tweets\_clean AS

SELECT

id,

ts,

text,

m.country

FROM tweets\_simple t LEFT OUTER JOIN time\_zone\_map m ON t.time\_zone = m.time\_zone;

Parsing the tweets contents into separate words

create view tweets\_explode as select id, words from tweets\_raw lateral view explode(sentences(lower(text))) dummy as words;

create view tweets\_explode\_words as select id, word from tweets\_explode lateral view explode( words ) dummy as word ;

create table right\_dictionary as select \* from dictionary where word <> "stark"

Dictionary uses word Stark as negative. But Tony Stark is the name of the char . Hence word is removed from new dictionary

create view tweets\_polarity\_pts as select

id,

tew.word,

case d.polarity

when 'negative' then -1

when 'positive' then 1

else 0 end as polarity

from tweets\_explode\_words tew left outer join right\_dictionary d on tew.word = d.word;

Based on tweet id, the sum of individual polarity words are done to get the overall sentiment of the tweet

create table tweets\_sentiment stored as orc as select

id,

case

when sum( polarity ) > 0 then 'positive'

when sum( polarity ) < 0 then 'negative'

else 'neutral' end as sentiment

from tweets\_polarity\_pts group by id;

CREATE TABLE tweetsbi

STORED AS ORC

AS

SELECT

t.\*,

case s.sentiment

when 'positive' then 2

when 'neutral' then 1

when 'negative' then 0

end as sentiment

FROM tweets\_clean t LEFT OUTER JOIN tweets\_sentiment s on t.id = s.id;