**Iron Man**

* We have the dataset located in HDFS. It contains the tweets.
* Polarity of common words is stored in a dictionary file in HDFS.
* Copy the /data/SentimentFiles to /user/rvakhari9232 in HDFS (We use -cp to copy file within HDFS. To copy files to or from local system, we use -copyToLocal or -copyFromLocal)

A screenshot of a social media post

Description automatically generated

1. **ADD JAR** hdfs:///data/hive/json-serde-1.1.9.9-Hive13-jar-with-dependencies.jar;

**SET** hive.support.sql11.reserved.keywords=false;

1. **CREATE DATABASE IF NOT EXISTS** rvakhari9232;

**USE** rvakhari9232;

1. Creating the external table **“tweets\_raw”.** It contains details of each tweet.

**CREATE EXTERNAL TABLE** tweets\_raw (

id **BIGINT**,

created\_at **STRING**,

source **STRING**,

favorited **BOOLEAN**,

retweet\_count **INT**,

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 **INT**,

month **INT**,

day **INT**,

hour **INT**)

**ROW FORMAT SERDE** 'org.openx.data.jsonserde.JsonSerDe'

**WITH SERDEPROPERTIES** ("ignore.malformed.json" = "true")

**LOCATION** '/user/rvakhari9232/SentimentFiles/SentimentFiles/upload/data/tweets\_raw';

1. Creating an external table **“dictionary”**

**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/rvakhari9232/SentimentFiles/SentimentFiles/upload/data/dictionary';

1. Creating an external table **“time\_zone\_map”**

**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/rvakhari9232/SentimentFiles/SentimentFiles/upload/data/time\_zone\_map';

1. Create the view **“tweets\_simple”.** It contains tweet id, the timestamp of tweet, tweet text and user's time zone.

**CREATE VIEW** tweets\_simple **AS**

**SELECT** id,

cast(from\_unixtime(unix\_timestamp(concat('2019',substring(created\_at,5,15)),'yyyyMMdd hh:mm:ss')) as **TIMESTAMP**) ts,

text,

user.time\_zone

**FROM** tweets\_raw;

1. Create the view **“tweets\_clean”.** It maps user's timezone to the country. Each row of the view contains tweet\_id, the timestamp of the tweet, tweet text and user's country (which is derived from time zone)

**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;

1. Create view **“l1”.** It converts each tweet into lower case and explodes it into a list of words.

**CREATE VIEW** l1 **AS**

**SELECT** id,

words

**FROM** tweets\_raw

**LATERAL VIEW** explode (sentences(lower(text))) dummy as words;

1. Create view **“l2”.** It stores every word of a tweet in a new row.

**CREATE VIEW** l2 **AS**

**SELECT** id,

word

**FROM** l1

**LATERAL VIEW** explode(words) dummy as word;

1. Create view **“l3”.** It joins the “l2” view with “dictionary” table and stores polarity of each word

**CREATE VIEW** l3 **AS**

**SELECT** id,

l2.word,

**CASE** d.polarity **WHEN** 'negative' **then** -1

**WHEN** 'positive' **then** 1

**ELSE** 0

**END AS** polarity

**FROM** l2

**LEFT OUTER JOIN** dictionary d

**ON** l2.word = d.word;

1. Create the table **“tweets\_sentiment”.** Each row of the table stores the sentiment of tweets.

**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** l3

**GROUP BY** id;

1. Create the table **“tweetsbi”**.

**CREATE TABLE** tweetsbi

**STORED AS ORC** **AS**

**SELECT** t.\*,

s.sentiment

**FROM** tweets\_clean t

**LEFT OUTER JOIN** tweets\_sentiment s

**ON** t.id = s.id;

Now we have a sentiment of each tweet along with country from where this tweet was tweeted.