Sentiment Analysis of Smart Devices Topics from Twitter Data

Allan Lo, Eric Yang, Prajakta Pandharkar

MS3: Progress Report (1-2 -pages): description of first component of the project idea summary and justification, a partial description of data acquisition and organizing strategy and justification, tools/third party libraries description usages and initial performance evaluation on the adopted data acquisition strategy.

Project Summary:

As described in our project proposal, social media platforms provide a tremendous opportunity for marketing professionals to discover customer and prospect sentiment analysis. Our plan is to analyze the smartphone related tweets for few major brands: Apple iPhone, Samsung Galaxy and Google Pixel.

Progress Summary:

We have started collecting data in batch mode from Twitter using the Twitter's Search API with keywords and hashtags. To date we have collect 50k raw tweets on the the three major smartphones we are intending to analyze. We've developed a script in python to aggregate the batch files in order to load them into Mongo DB. We intend to use Mongo DB as our data lake where we will store raw data. From Mongo DB we have developed a script to transform the data in preparation to be loaded into HDFS/Hive via a python script. Some examples of the transformations being performed by the script include, reducing tweets down to unique users, removing duplicate tweets, removing retweets, etc.

Data Collection:

We narrowed down the tweets related to only a few branded devices in search query option in the python script in order to reduce the noise and volume of tweets.

We experimented options to collect tweets using near real time streaming API or making a batch process on Twitter search API that runs once per day using keywords. We did not see huge volume of data triggered by any particular event anytime we decided to pursue the batch method to collect the tweets. We are collecting tweets in. json format using the python tweepy library in python scripts as shown in github path. At present the script records the time required to collect the required tweets and displays messages in case the threshold in time is breached.

Further enhancements will focus on improving connectivity loss for any reason and handling exceptions in program errors.

Data Cleaning:

At present the tweets contain various duplicate tweets as well as tweets that contain homonyms or non-smartphone context. We are planning to clean such data using the python script which will take focus on de- duplicating data, removal of retweets and retention of only relevant smartphone brands.

Architecture:

Our initial plan was to make a data lake with ELT architecture. As previously mentioned, we're utilizing python scripts with the Tweepy library to extract the data. We plan to use MongoDB NoSQL db for the ease of storing large volume of tweets in .json format. We are still working on feasibility of the solution with MongoDB in the mix.

For now, we plan to use Amazon instance to host MongoDB for raw tweets storage and then apply various data python cleaning scripts to create .csv files of selected tweets, attributes such as text, entities etc and add transformed attributes useful for sentiment analysis. These files will then be aggregated and transformed further to include pre calculated attributes useful for statistical analysis and presentation in the reporting layer. For performance gain pre-calculated attributes and transformed data in this fashion will be loaded in Hive db.

Finally, the dashboard and analytical reports will be presented in Tableau. In these reports we will compare and contrast sentiment analysis of smartphones based on locations, features etc.

Sentiment Analysis Process

As we progress further, we are working on exploring Nltk library for text analytics and categorize tweets corresponding to sentiments. As part of basic sentiment analysis, we score the tweets using tokenization based on text used in the tweets. Words such as “great”, “good”, “luxurious” etc. will be scored as positive tweets whereas any “terrible”, “sucks” etc. will be scored negative.”