George Akritidis
Department of Media Technology
Linnaeus University
ga222ey@student.lnu.se

Introduction

This document describes the final project for the course Programming for Digital Humanities. The final project was a combination of the knowledge we acquired in the previous assignments with the addition of some new features such as sentiment analysis. The topic and the content of the final project was free to students, so I selected something that I am very keen on. Since I am a fun of science fiction movies, I thought that it would be a good idea to make a project that is related to the latest star wars movie, 'The Last Jedi'. Specifically, my idea was to collect a certain amount of tweets regarding this movie and make a sentiment analysis on them. Then, I created text files with the values of the findings and finally I created charts to visualize the numbers I found. So, basically my project involves three main categories of digital programming which are text processing, statistics and visualization, and sentiment analysis. From the outcome of the program that I implemented the reader will be able to examine whether people share a positive, neutral or negative opinion about the movie. What is more, the reader will be able to read the actual comments that were written from the twitter users.

For the project I have upload a video of the presentation of my work in YouTube. The url is:
https://www.youtube.com/watch?v=cQCsa2yqxZw&feature=youtu.be

Approach to solving the given problem and tasks

First of all, I made a thorough research on what technologies are being used to import tweets and how they can be stored. Quickly, I came to the decision that the tool Tweepy is the most suitable for my case. Tweepy is an easy-to-use Python library for accessing the Twitter API and it supports oauth authentication. A prerequisite to use Tweepy library is to create a twitter application through the website https://apps.twitter.com/, so that you can acquire a consumer key, a consumer secret, an access token and an access token secret. Then, I studied the ways you can import tweets, because there are different methods to do the importing. For example, I had to decide in the beginning whether the tweets that would be imported would be on real-time or simply random tweets from the last 7 days. I concluded that in my case, importing tweets randomly would be the most suitable solution; since it doesn't

effect the outcome in any way and it is easier.

Furthermore, I set the search word to be 'The last jedi', so that twitter API searches for tweets that are specifically written for this movie and not for any previous one. Since, the twitter API only imports 100 tweets at a time, I set the maximum number of tweets to 100. Also, for the language parameter I set it to english for obvious reasons and I didn't set any specific geolocation coordinates as I thought that it would not matter to the final outcome.

Now, regarding the actual comments, I decided not to store them in a json file but simply to display them in the console after the execution. The viewer can read the text one by one, in the result window of the program Sublime Text. The Twitter API has the limitation to display only 100 times at one search from the last 7 days, that is why even if I set the parameter max_tweets to another higher number the API will display the same tweets over and over again.

As far as the sentiment analysis is concerned, I followed the guidelines of the instructors of the course and I made use of the library TextBlob. Sentiment analysis, otherwise known as opinion mining is the process of determining the emotional tone behind a series of words, used to gain an understanding of the the attitudes, opinions and emotions expressed within an online mention. TextBlob is a Python library for processing textual data and its main property is sentiment. Sentiment contains two variables in a form of tuple, polarity and subjectivity. Polarity takes values in the range [-1, 0, 1], where -1 is the most negative, 0 means neutral, and 1 means very positive. Polarity expresses the feeling of someone about something. Subjectivity takes values in the range [0, 1] and it indicates whether (according to the classifier) the text is subjective or objective. Values that are close to 0 indicate that the text in objective, whereas values close to 1 indicative objectiveness.

After building the code, four text files are generated. The first ('result.csv') contains the values of polarity and subjective from all the tweets. The second ('polaritylist.txt') holds the polarity values and the third ('subjectivity.txt') holds the subjectivity values. The fourth text file ('urls.txt') contains the urls that were included inside the comments.

To give a better view of the findings of my project, I visualize my results by creating two charts. The way of doing this, is by using the second and third text file with Excel. I create two bar charts which depict which values occur more often.

Outcomes/Analysis of results

The results are different every time I execute the code. That's because the API imports different tweets each time it searches the web for comments related to the key phrase. Despite this fact, the mean value of polarity and subjectivity are more or less the same each time. The mean value of polarity is steadily

around 0.10 and the mean value of subjectivity is between 0.20 - 0.30. This indicates that the opinion of the people about the movie is relatively neutral. It gives me the feeling that this movie didn't make a good impression to the audience, but at the same time it didn't create too much negative impression.

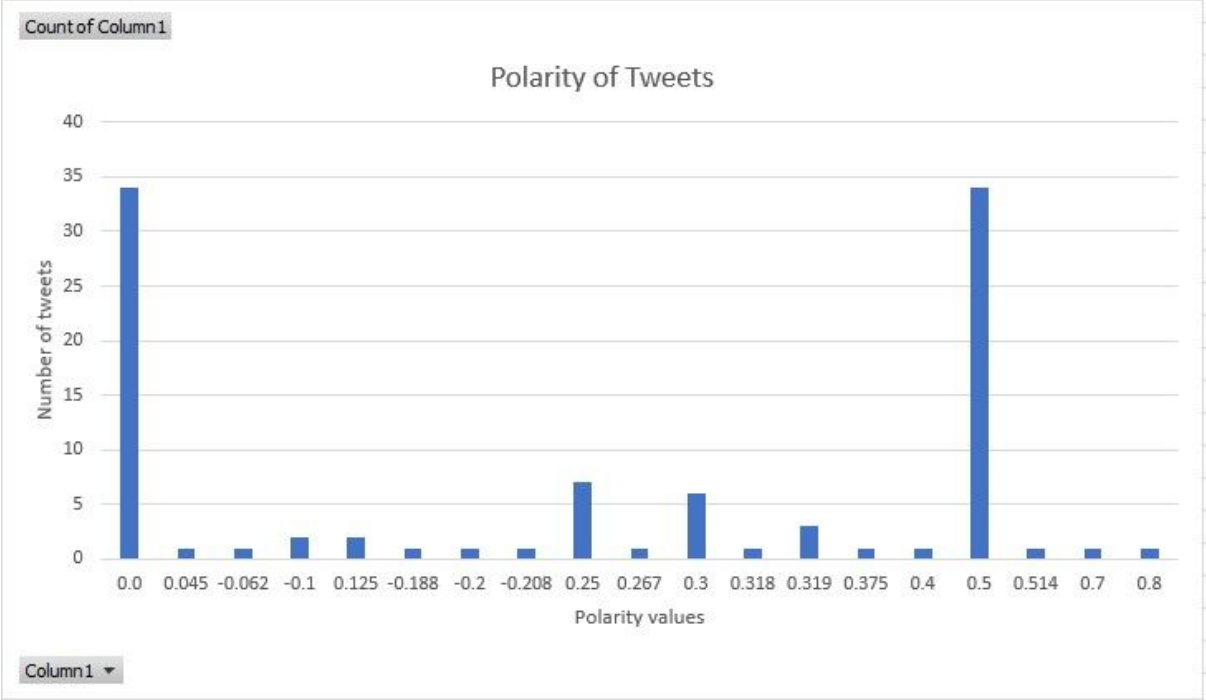
I noticed in the results that the Twitter API has one major flow. Since, it can only import 100 tweets, some of them tend to be same; either from the same twitter user or the same tweet shared by another user. Yet, I believe still we get a clear of the public opinion about the movie.

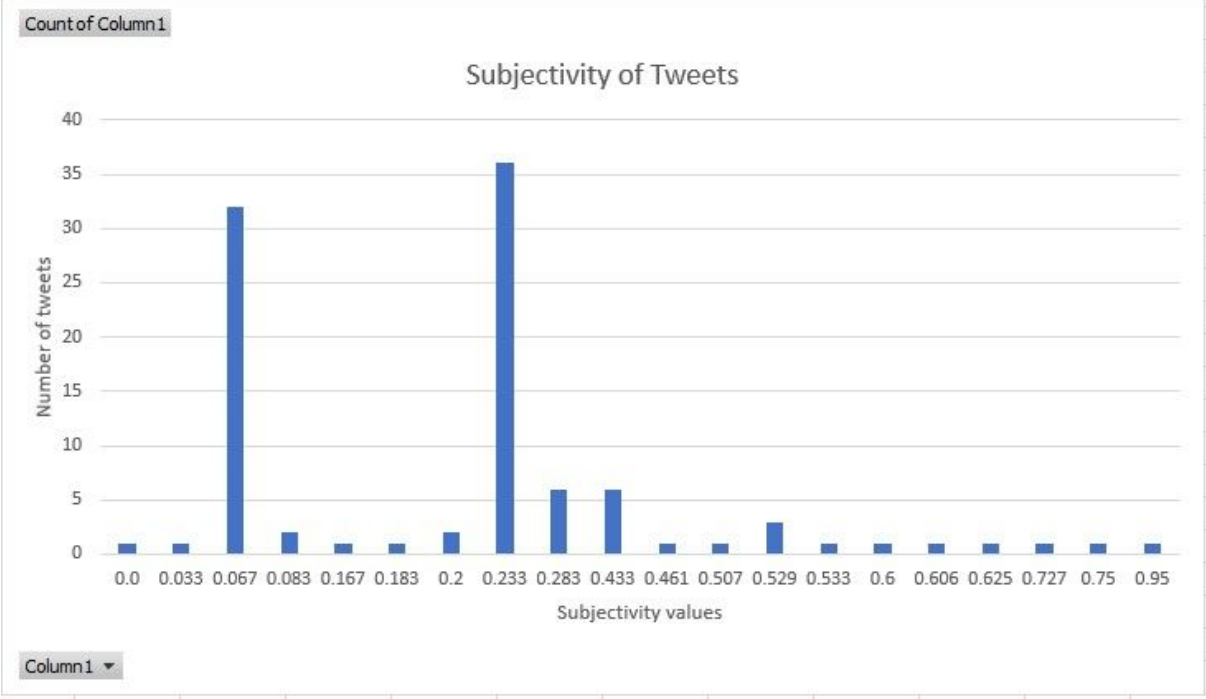
Conclusions and Reflections

The final assignment for this course was rather challenging. The project was more difficult to be implemented as I initially estimated in the project proposal. What troubled me, in particular, was the selection of the most suitable chart to visualize the results. In the beginning, I made great effort to generate charts with the use of Python and specifically by using the Plot library. Unfortunately, very soon I found myself in a dead end as I wasn't sure what values I should put in the x and y axis. So, I decided that it is easier to simply create text and csv files and later create the charts through Excel.

The idea for this project came from my curiosity to discover, if the excessive promotion and good critics that the movie received, actually correspond to reality. It proved to be, considerably, away from the truth, as I observed in the comments that people were not impressed after all so much by the last star wars movie. I realized after the completion of the project, what capabilities a common programmer has by using programming technologies. This knowledge that we acquired can be used to inform the public about misconceptions and disinformation, that can be happening intentionally some times by the media.

Appendix


Count of Column1
Polarity of Tweets
Number of tweets
40
35
30
25
20
15
10
5
0
0.0 0.045 -0.062 -0.1 0.125 -0.188 -0.2 -0.208 0.25 0.267 0.3 0.318 0.319 0.375 0.4 0.5 0.514 0.7 0.8
Polarity values
Column1


Count of Column1
Subjectivity of Tweets
Number of tweets
40
35
30
25
20
15
10
5
0
0.0 0.033 0.067 0.083 0.167 0.183 0.2 0.233 0.283 0.433 0.461 0.507 0.529 0.533 0.6 0.606 0.625 0.727 0.75 0.95
Subjectivity values
Column1