

# Analyzing Facebook Friends

Shrinidhi Thirumalai, Elizabeth O'Callaghan

## 1 Deliverable One

**Project Overview:** For this project, our goal is to order a person's facebook friend list from most positive to negative sentiment as well as most subjective to most objective. To do this, we first get all the posts related to each friend. We then filter the posts to only include ones that have been written by the person. Once the posts have been filtered, they're saved into a dictionary with the key as the friend and the value as a string of all the person's statuses concatenated together. These are then ordered both by positivity and subjectivity, then returned to the user.

**Implementation:** **Question:** How does your code work? What libraries did you use? How would someone run your code? What data structures(lists, dictionaries) did you use in your program and why?

**Libraries:** Our code makes use of the libraries `pattern.web` and `pattern.en`. `pattern.web` was used to gather our data while `pattern.en` was used to analyze our data. Firstly, `pattern.web` searches facebook for different relevant fields, such as news(used to return the posts related to each facebook friend). We then filtered this news through other functions included in the library that provided us with more information about each post. For example, we filtered posts by author, text, and ID. `Pattern.en` was then used to get the sentiment and objectivity negative one to positive one.

**Functions:** The functions we used are listed below, which each of their inputs and outputs explained. To run our code, a person would run the main function script, which calls all the unit tests and the facebook code. This allows the user to not only see an order of most to least sentiment and objectivity, but also shows the "winner" of each category.

**Data Structures:** We mostly used dictionaries in our code rather than lists. We did so because we wanted to not only create a set of values but also wanted to back out which friend corresponded to each of these values. By using dictionaries, we could easily do so instead of having to use multiple for loops. The only exception to this are the functions which sort

**Results:** It was very interesting to find our most positive, negative, subjective, and objective friends, and look at correlations between positivity and subjectivity. Given more time it would have been interesting to produce a graph with the y axis as subjectivity and the x axis as positivity.

**Reflection:** We divided the scope of the project very well. We designed code that allowed Shrinidhi to work on the facebook interfacing, while Lizzy worked on analyzing the data. Originally we had hoped to build a map of which of our friends talked most similarly; however this proved not to be possible for us given other work. We thus did work analyzing positivity and subjectivity which proved to be an obtainable goal. We were able to unit test very well for the analyzing part of the code.