Final Project-

Analyzing Yonsei Bamboo Facebook page through Python

2012112082 Myeongjin Ko

2012121137 Kakyung Kim

2015121183 Hyungjun Ha

2015123072 Chaewon Park

Index

1. Introduction

First, we wanted to analyze people's emotional changes in a statistical way. And we thought that we can find it through Yonsei Bamboo Facebook Page because it is the space that people can report their opinions or thoughts anonymously. So people can post anything without any constraints.

Also there are several administrators of Yonsei Bamboo Facebook Page. Their main job is to sort the reports and post them on the page. In this process, there is a principle that the report that people send should be posted within 72 hours. We wondered if the administrators abide by the rules, so we were also determined to check it.

2. Data Crawling

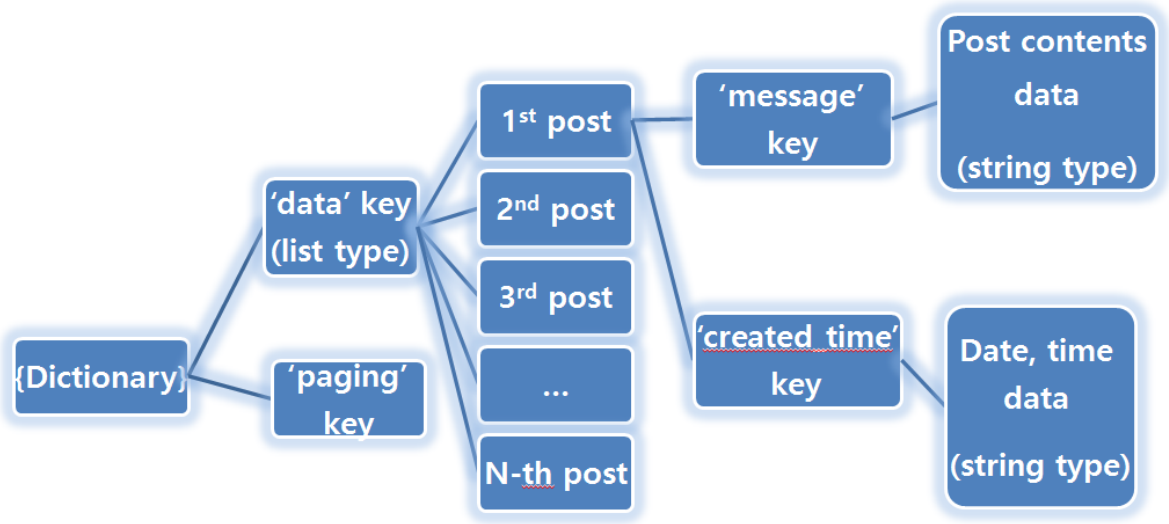
1. Graph API

- API is a basic way to approach data on a website. Facebook provides "Graph API" and we can get Facebook data such as posting texts, number of likes, or created time, also can upload a new post on my timeline using it.
- We used Graph API to get yonsei bamboo facebook data and Python provides 'facebook-sdk' library which is designed to support the Facebook Graph API
(facebook library download link https://pypi.python.org/pypi/facebook-sdk)

2. Access Token
- We need a token to get access to facebook data through Graph API and it is renewed every two hours. (getting token link https://developers.facebook.com/tools/explorer/145634995501895)

3. Data Description
- When we first crawl the data it comes as a dictionary type, containing two keys, 'data' and 'paging'. Our targeting data such as post messages and post created time is in the 'data' key.
- 'Data' key value has a list type data
- Length of this list equals to the number of posts we crawled, hence, THIS_LIST[i] can give us the i-th post data
- There are other dictionaries again and key values are 'message' and 'created_time'. 'Message' key has text contents of the post and 'created_time' has time when the post was uploaded. We made a new blank list and put these two data as a string type
(ex- ['Hi everyone', '2016-06-02 T04:27:24+0000'])

- At last, we wrote these lists by row as a csv file using csv library



3. Re-organizing the Data

We took the CSV file back, which contains the contents of Yonsei Bamboo facebook Page as string files (Got about 3266 posts). We read this file line by line and stored each line as a component of a list. And we input three words that the users want to find. When storing the line as a list, we used the 'split' function so that we can identify a word which makes us recognize the reported time(Not created time).

And we made a list which has [reported time, counting number] of each post. Then we append each list to a large list. And we made a Dictionary, created the key for every alternative hour, and the value as frequency.

First, we set up the value as zero. And with the Data list, we sorted the first component of the list as the key of Dictionary and added the second component of list(frequency of word) as the value of Dictionary. So we finally made a Dictionary which has the format of [every alternative hour, frequency of the input words] (ex – ['오전11시', 30]).

However, we found that the total number of post at dawn has lower number than other times. So we thought that if we want to find more meaningful value, we should find the ratio of the input words. We repeated the same process and made the Dictionary of total number of post, and a Dictionary of Ratio(Frequency of Words/Total number of Posts).

4. Visualization

To find the correlation between number of words appeared in 'Yonsei Bamboo Facebook Page' posts per time, we used matplotlib. Matplotlib offers various functions to visualize data filed in lists, dictionaries and etc. Among the functions we choosed pyplot to set basic facts that constitutes a graph.

Before putting data into pyplot function, we had to switch key value in existing dictionary data_words and data_ratio because the key values in those dictionaries were written in Korean. So, we set new dictionaries, new_dic and new_dic1 then switched '오전' to 'am', and '오후' to 'pm'.

Then we set x axis as time and y axis as ratio for first graph, 'Ratio of words appeared per Time' and for second graph, 'Number of words appeared per Time'. To label time we set new dictionary, 'time' and 'time1' then put strings from 'AM12 to PM11' in xticks function. By using bar function, we could create bar-type graph from data in dictionaries in new_dic and new_dic1. Also, to make comparison between ratio and absolute frequency easier, we used function subplot to show the graphs of ratio and number in a screen. Finally, we decorated the graph by using style.use('ggplot').

5. Fun Facts of Bamboo Posting

We also tried to create an algorithm for a program to calculate the time it takes for a Bamboo Facebook Page manager to upload a post after it is written by a Facebook user. We wanted to find the average, maximum, and minimum uploading time. Furthermore, since it is stated in the Bamboo Page policy that all posts are uploaded in 72 hours, we wanted to verify if this was true or not.

For the first step, we imported the csv file that holds the data of the post. We opened the file and saved the information in a list called 'data'. We also created a spare list called 'time_data' for later use. The variables 'violate', 'count', 'total_how_long', and 'k' is initialized as 0.

Now, we tried to extract the time when the post was written from each post. Each post is located in data[k][0], so we used a while loop with k being smaller than the length of the data, or the total number of posts. We sliced the post as data[k][0][18:41] and stripped away any meaningless blank spaces using the strip function. Before printing out the time when the post had been written, we discovered that some users made requests that their written time be erased for privacy matters. In these cases, it is impossible to calculate the time gap, so we decided to ignore

such posts. To determine whether the user had requested time eradication, we checked whether the string '2016' is in 'time_written'. If '2016' is present, then the time eradication has not been requested. If not, time eradication has been requested and therefore we had to exclude that particular post.

With the 'time_written', we sliced it again to get the month, day, hour, minute, and seconds. However, we discovered that the length of 'time_written' varied since the day and hour value may be a one digit number like '7' or a two digit number like '21'. So before slicing it, we needed to classify it into three cases- when the length is 23, 22, and 21. There were cases when unnecessary blanks were included in the sliced string so we stripped them off as well.

In order to calculate the time gap easily, we needed to convert the hour values of time_written into a 24- hour system. It's because in the csv file, the time_written was given in a 12 hour format and the time_uploaded was given in a 24 hour format. Hence, if the string '오후' was in time_written and the hour value was from 1 to 11, we added 12 to it. If the string '오전' was in time_written and the hour value is 12, we subtract 12 from it.

The time when the post was uploaded by the Bamboo Facebook page can be organized by replacing the '-' with '.' for coherence with the time_written. Slicing was also needed here to get the month, day, hour, minute, and second value.

Now, in order to calculate the time difference, we converted the month, day, hour, minute, and seconds of time_written and time_uploaded into seconds. To elucidate, we subtracted 1 from the month value in order to get the value of previous month. Then we added up the number of days from January to get an accumulated value of days. Then, we added the day, hour, minute, and second value of time_written converted into seconds to get the number of seconds that passed since the start of January 1st. Then we subtracted the 'total seconds of the written time' from the 'total number of seconds of the uploaded time'. Plus, since the time_uploaded was expressed in Greenwich Mean Time, we had to add 9 hours in order to change it into Korean Standard Time. Thus we got the time gap of 'how_long'.

We also added that value to the list called 'time_data' and used max() / min() method to find the max and min values. We also added each value to total_how_long to calculate the average time gap. We used the try-exception method because there were cases when an exception occurred because the format of the post was unanalyzable. We also counted the number of posts that have been uploaded after 72 hours.

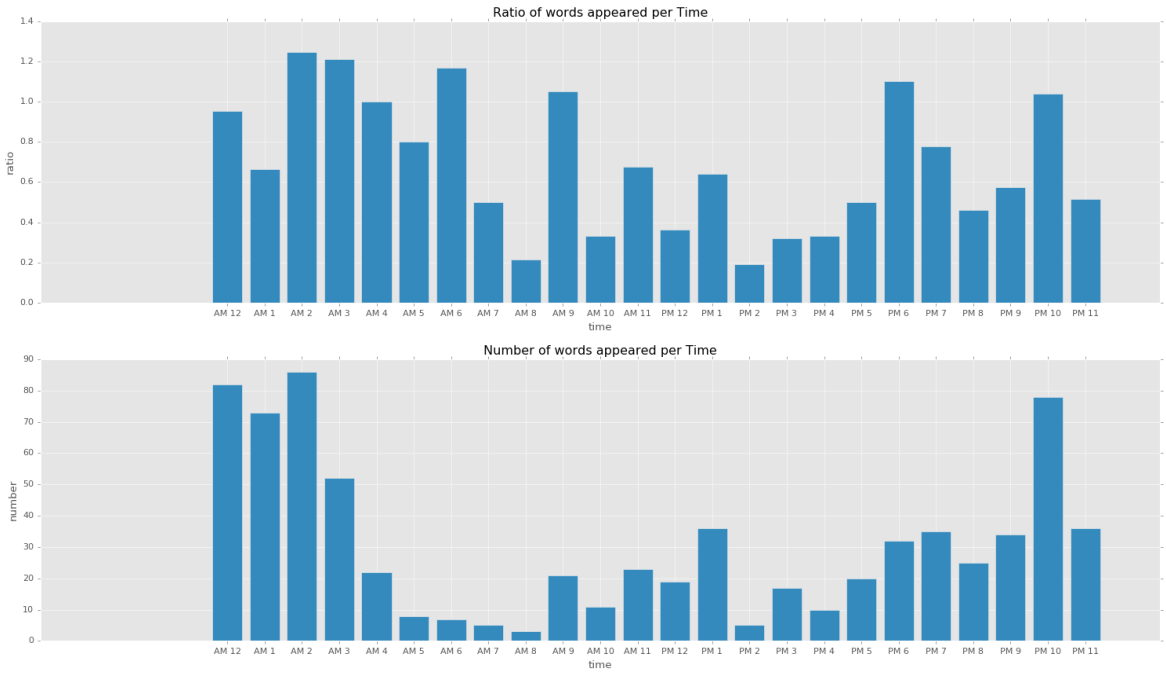
Hence we got the following result:

On average, the post is uploaded in 1 day(s) 2 hour(s) 21 minute(s) 58 second(s)

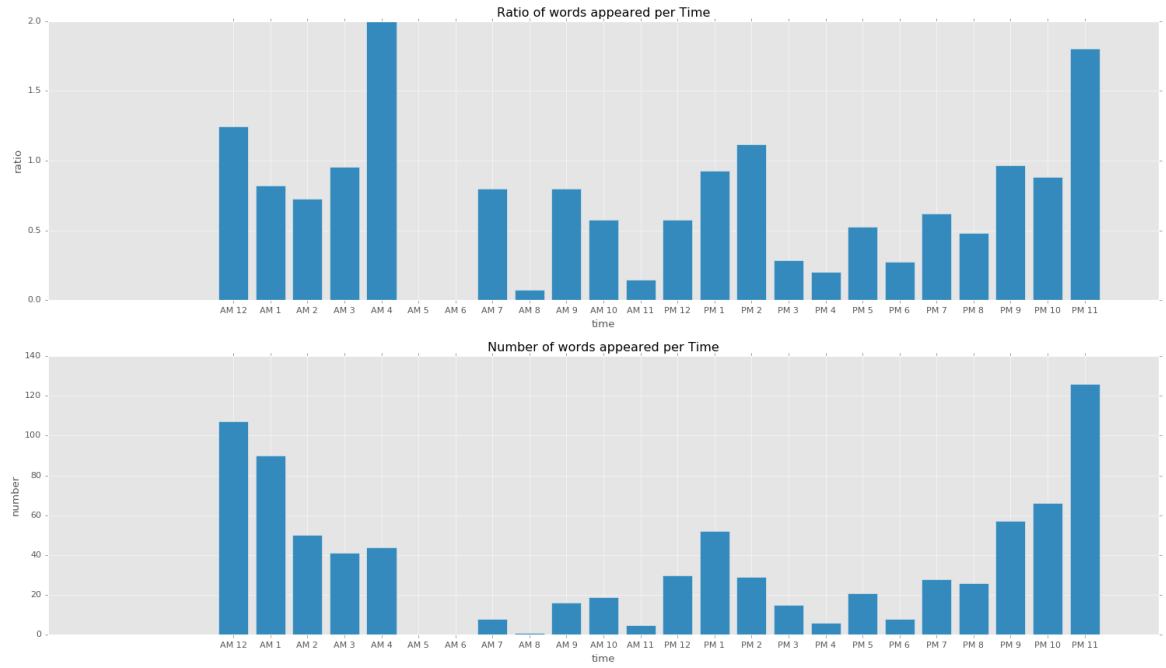
The maximum amount of time taken for a post to be uploaded is 3 day(s) 16 hour(s) 37 minute(s) 11 second(s)
The minimum amount of time taken for a post to be uploaded is 0 day(s) 0 hour(s) 2 minute(s) 23 second(s)
The Bamboo Forest promises that each post will be uploaded in 72 hours. Of the total 3060 posts, 42 posts have violated this regulation.

6. Conclusion

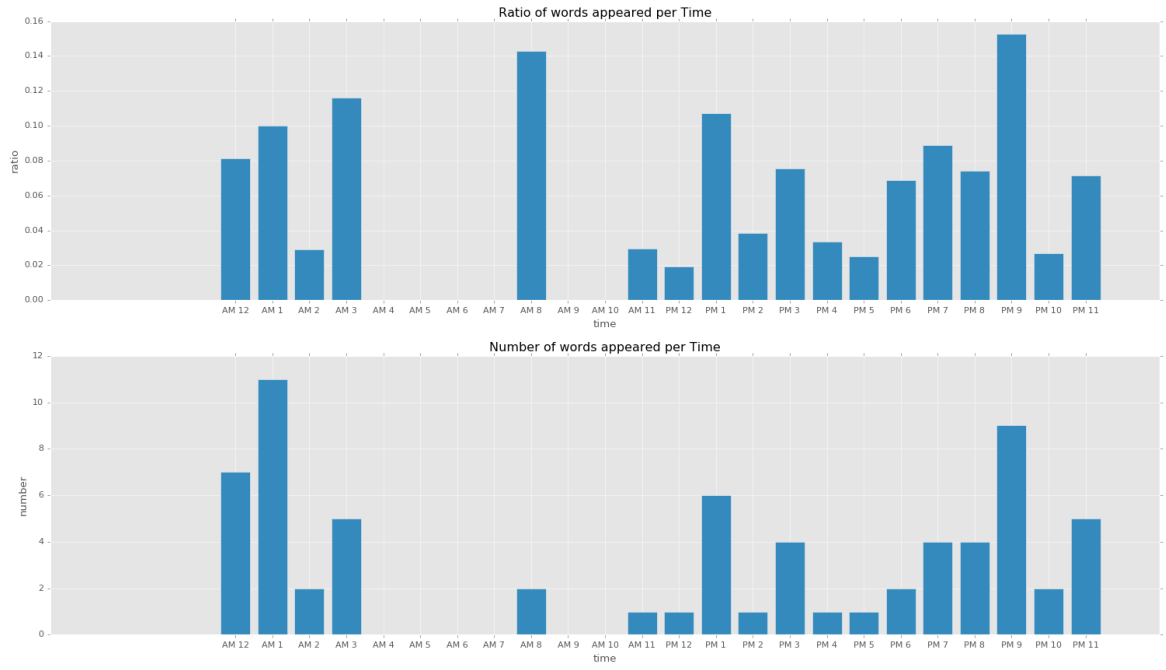
In conclusion, we found that people become emotional at night and dawn time. We first input the three words '사랑', '썸', '연애' which is about the feelings of love. And we found that it has the highest rate(about 1.25) at 2AM and the second highest rate(1.21) at 3AM. And also generally high in dawn time.



And then we put '아빠', '엄마', '집' which is about family. Also we could find that generally it has high rate in dawn time, especially in 4AM when the rate reach 2.0. Through these data, we found that people generally become emotional at the dark time of day(After the sun sets).



Also, we put '짜증', '빡', '어이없' which means irritating and anger. Especially 9PM had high rate of the word which reach about 1.5. Also the second highest rate was 8AM when the rate was about 0.14. We interpret that this is the time when students usually do their homework. So we guess that many students who are doing their homeworks may report their feeling to Yonsei Bamboo Facebook Page where they can say anything anonymously.



7. Roles of Each Member

Kakyung Kim- Researching Facebook Graph API, Took a leading role in Writing code to Crawl the Yonsei Facebook Page Data, Re-Oraganizing the raw data, Writing 'Data Crawling' part of report and ppt, making the video file of process. Arrange the powerpoint.

Chaewon Park- Researching Facebook Graph API, Writing code to measure the time gap, Writing 'Fun Facts of Bamboo Posting' part of report and ppt, making the video file of process, Edited the template images for the ppt, Arrange the powerpoint.

Hyungjun Ha- Took a leading role in researching Facebook Graph API, Writing code to Crawl the Yonsei Facebook Page Data, Visualizing the Crawled data as graph, Writing 'Introduction', 'Re-organizing the Data', and 'Conclusion' part of report and ppt, Arrange the Final report

Myeongjin Ko- Researching Facebook Graph API, Took a leading role in Visualizing the Crawled data as graph, Writing 'Visualization' part of report and ppt, Arrange the Final video file.

8. Used Libraries

import facebook

import csv

import matplolib

import requests