BIA-660 Midterm Project

Analytics & Comparison of Apple Products with Twitter

By:
Colin Mills
Qiang Tong
Xiangda Shi
Xin Gao

Instructor:
Prof. Winter Mason

Questions

- Which Apple mobile OS is more popular? iOS4 ~ iOS7.

- While version of iPhone is more popular? iPhone 3GS ~ iPhone 5S.

- iPhone 5S vs. iPhone 5C?

3 questions about our project

Headlines

- Data Collection

- Data Analysis

- Data Visualization

3 main topics

Data Collection

- Why don't we use Twitter API?

- Twitter search URL:
  http://twitter.com/search?q=

- topsy.com/tweets search URL:

Twitter API cannot get tweets more than a week ago.

Data Collection

- topsy.com/tweets search URL:

http://topsy.com/s?type=tweet&q=
iOS%204&language=en&offset=00&mintime=1277078425&maxtime=1308700814

| iOS%204 | & | language=en | & | offset=00 | & | mintime=# | & | maxtime=# |

"iOS 4"

"pages"
00, 10, 20

1277078425?
timestamp.
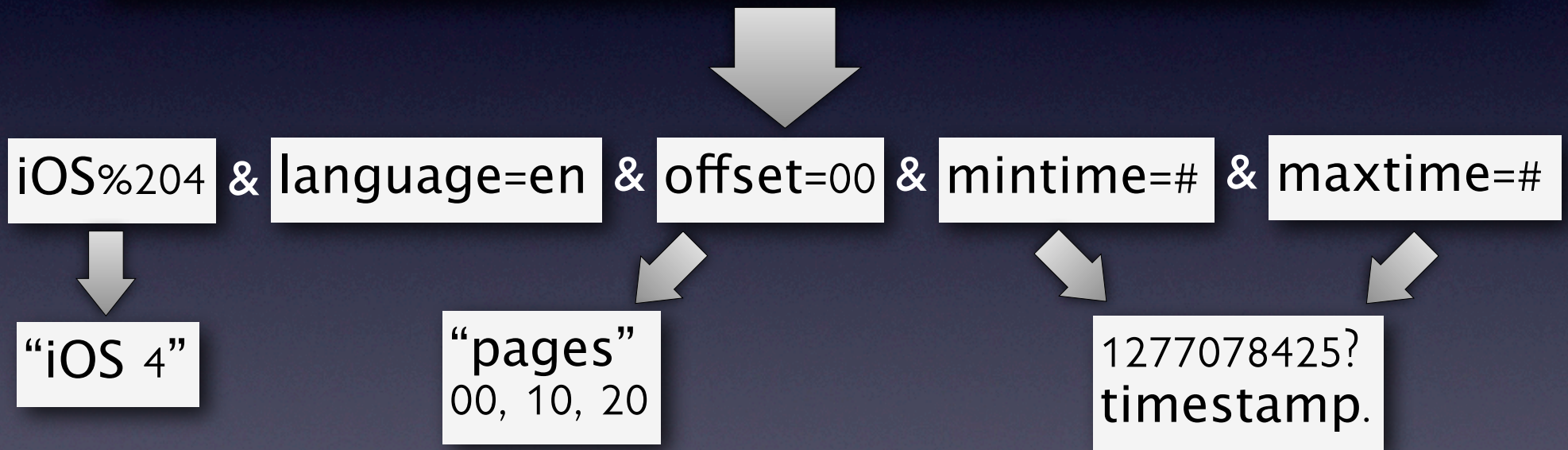
- Another problem, Javascript?

URL explanation.

Data Collection

- "Selenium" to the rescue!

HTML Code

```
<div class="media-body">
   <h5 class="media-heading">
      <a href="http://user_id_url">
         Mashable
      </a>
   </h5>
   <div>iOS 4 Has Arrived <!--Content--></div>
   <ul class="inline">
      <li>
      <a class="muted" herf="http://tweet_url">
      </li>
      <li><!--"Reply", "Forward"...--></li>
   </ul>
</div>
```
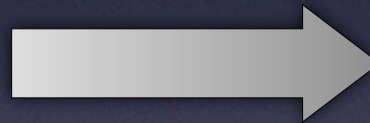
Selenium
BeautifulSoup

MySQL DB

iphone4_tweets

- tw_url: VARCHAR(255)
- usr_id: VARCHAR(45)
- usr_name: VARCHAR(255)
- usr_url: VARCHAR(255)
- tw_content: VARCHAR(150)
- tw_time: DATETIME

- More than 2,200 tweets each keyword.

Using tools of Selenium & BeautifulSoup to collect and parse data from webpage, then store the data into database.
I hope to make a demo about how "Selenium" works. Run a small demo code.
We managed to a large amount of data.

Data Analysis

- "Time" matters.

- Retrieve tweet content from database.

- Positive or Negative?

- Special Dictionary.

Pos_words.txt

```
adorable accepted acclaimed
accomplishment achievement active
admire affluent amazing
...
...
wow yummy zeal zealous
```

neg_words.txt

```
abysmal adverse alarming angry annoy
anxious apathy appalling atrocious
awful bad banal barbed belligerent
...
,,,
woeful worthless wound yell yucky
```

Time matters:
We chose to collect the tweets when the iPhone or iOS was release. About 4 days before and after the official release date. This reflects more about how people thought about the product/software. Also, it's easier to get large amount of data.

Data Analysis

Sample Code:

```
conn=pymysql.connect('127.0.0.1',port,db_user,db_password,db_name)
    str_sql_count='SELECT COUNT(*) FROM '+tbl_name
    str_sql_get= 'SELECT TW_URL, TW_CONTENT FROM '+tble_name'
    cur = conn.cursor()
    cur.execute(str_sql_count)
    (all_count,) = cur.fetchone()
    list_pos = []
    list_neg = []
    cur.execute(str_sql_get)
    for row in cur:
        content = row[1].lower()
        for word in neg_words:
            if word in content:
                list_neg.append(content)
                break
        if word == neg_words[-1]:
            for word in pos_words:
                if word in content:
                    list_pos.append(content)
                    break
    cur.close()
    conn.close()
```

Sample code, also the key algorithm of checking if the tweet is positive or negative.
Briefly explain how this works:
Before this code, we have read the .txt files of positive or negative and save them into two lists.
Open a database connection;
run SQL query to get number of all tweets and

Data Visualization

positive/negative:

```
iPhone 3GS
Positive:  1048 / 3091    33.90489 %
Negative:   554 / 3091    17.92300 %
Positive/Negative:        1.89170
===================================
iPhone 4:
Positive:   707 / 3027    23.35646 %
Negative:   571 / 3027    18.86356 %
Positive/Negative:        1.23818
===================================
iPhone 4S:
Positive:   688 / 2758    24.94561 %
Negative:   554 / 2758    20.08702 %
Positive/Negative:        1.24188
===================================
iPhone 5:
Positive:   633 / 2492    25.40128 %
Negative:   552 / 2492    22.15088 %
Positive/Negative:        1.14674
===================================
iPhone 5S:
Positive:   615 / 2225    27.64045 %
Negative:   422 / 2225    18.96629 %
Positive/Negative:        1.45735
```

```
iOS 4:
Positive:   745 / 2552    29.19279 %
Negative:   530 / 2552    20.76803 %
Positive vs. Negative:    1.40566
===================================
iOS 5:
Positive:   787 / 2491    31.59374 %
Negative:   451 / 2491    18.10518 %
Positive vs. Negative:    1.74501
===================================
iOS 6:
Positive:   843 / 2611    32.28648 %
Negative:   492 / 2611    18.84336 %
Positive vs. Negative:    1.71341
===================================
iOS 7:
Positive:   729 / 2374    30.70767 %
Negative:   595 / 2374    25.06318 %
Positive vs. Negative:    1.22521
```

```
iPhone 5S Only:
Positive:   565 / 2207    25.60036 %
Negative:   478 / 2207    21.65836 %
Positive vs. Negative:    1.18201
===================================
iPhone 5C Only:
Positive:   619 / 2234    25.29096 %
Negative:   352 / 2234    15.75649 %
Positive vs. Negative:    1.75852
```
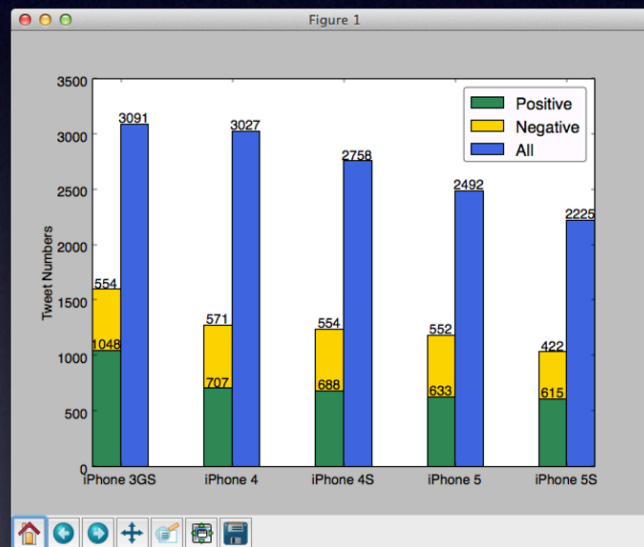
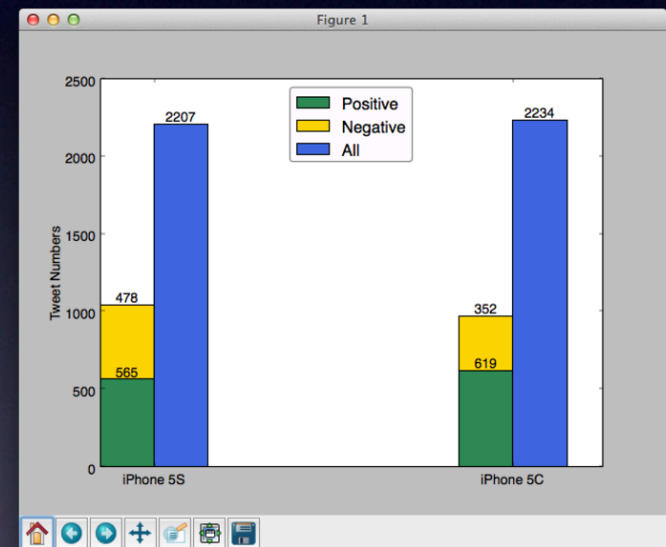Old school way, print out the data.

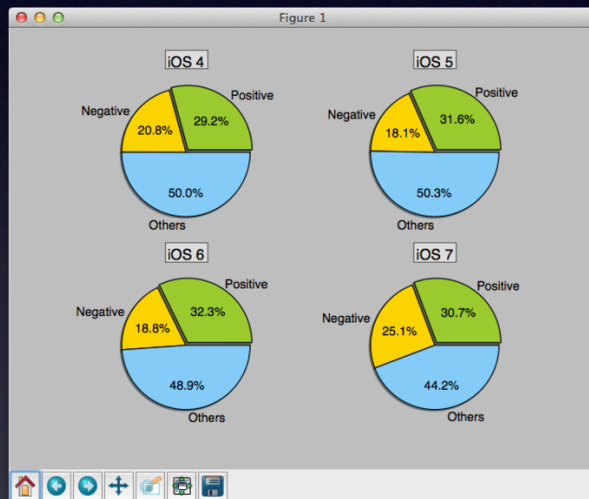# Data Visualization

Matplotlib - bar charts
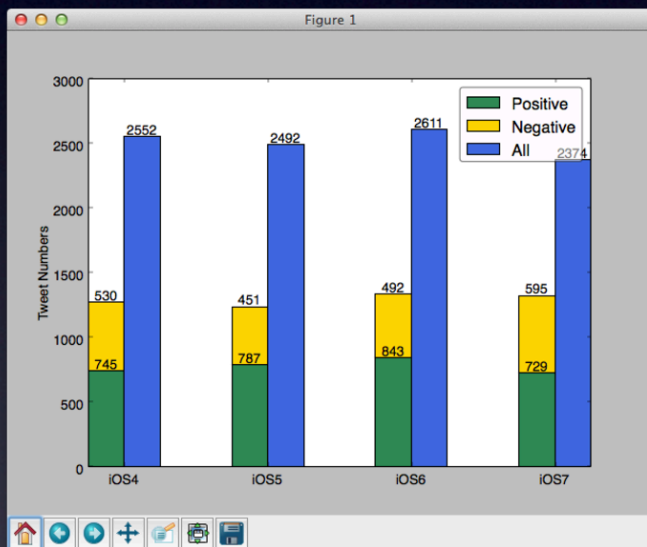
iOS 4/5/6/7   iPhone 3GS/4/4S/5/5S   iPhone 5S/5C

Data Visualization

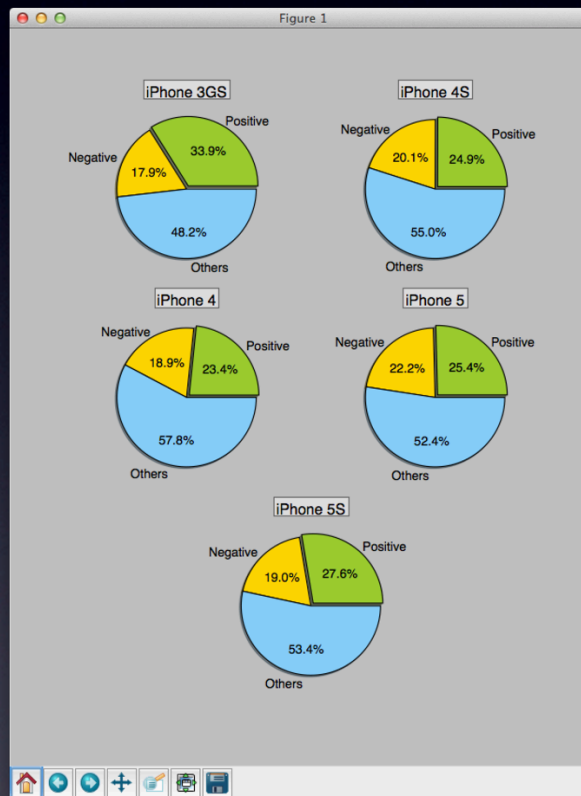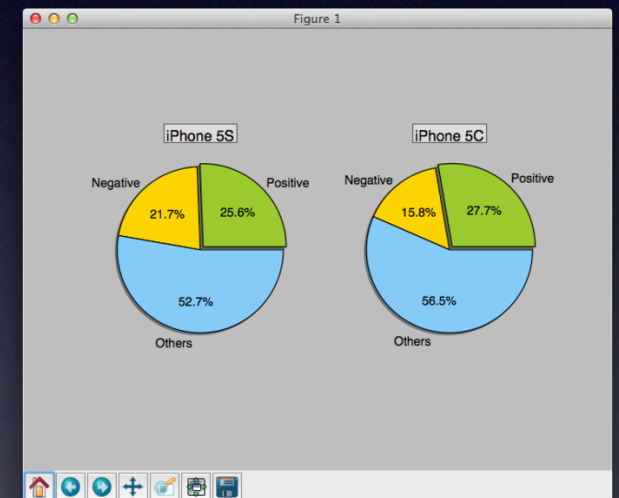## Matplotlib - pie charts

iOS 4/5/6/7

iPhone 3GS/4/4S/5/5S

iPhone 5S/5C

Conclusions

Q&A
Thank you!