Do like-minded People Go To Similar Places?
--Look Into Users' Foursquare Checkins and Twitter Followings

Ni Yan - SI601 Final Project Report

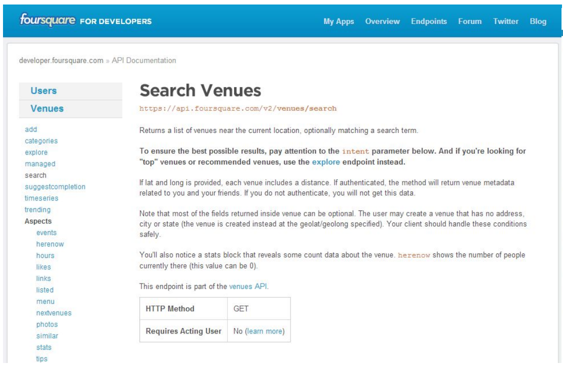
Motivation

As a member of the UM anime club "Animania", I sometimes go to the "poster party" to post fliers and posters on kiosks or restaurants around the campus. It is very interesting that the club ask us to visit Japanese and Chinese restaurants because they think people who dine there are more likely to love Japanese or Chinese food, and they are more likely to love Japanese or Asian culture, including anime and manga. Is it true? Do like-minded people go to similar places? I'm curious about the relationship between the place people go to and their interests. Now more and more people are using smart-phones to check in on Foursquare when they are in restaurants, bars, museums, airports and so on. They can also link other social media such as Facebook and Twitter. So if people check in a particular place, are there any common interests (e.g. The type of users they are following on Twitter) between them? Can I find some patterns and characteristics for a place? I want to find answers towards these questions, and evaluate whether posting fliers in Japanese restaurant can better target anime or manga lovers.

Data Source

I have used Foursquare API (https://developer.foursquare.com/docs/venues/search) (See figure 1) to search for venues that meet a search criteria. After entering the variables, the JSON results are shown in a beautiful structure on the web page, but the whole url is also given with OAuth token automatically set. The collected data is written into SQL for further manipulation. I also used the Tweepy, a Twitter API library for Python (See figure 2) (http://pythonhosted.org/tweepy/html/). Since Tweepy mirrors the Twitter API (https://dev.twitter.com/docs/platform-objects/users), I can refer the Twitter API filed-guide to get data using Tweepy (e.g.: "api.get_user.description").

There are several challenges using the data source. Firstly, only venue owners can get the whole list of checked-in users in the venue, and I can only get user information of "mayors" and users who left tips. Foursquare API only returns 20 most popular tips of a venue, the small amount of number of users cannot represent the real visitor population. Secondly, Twitter API set rate-limit of 150 requests per hour for third-party apps. In order to avoid hit the limit, I decided to only pull description of at most 10 twitter users and at most 10 of their friends for each venue. And I also use multiple accounts to switch from on and another.



Search Venues

https://api.foursquare.com/v2/venues/search

Returns a list of venues near the current location, optionally matching a search term.

To ensure the best possible results, pay attention to the intent parameter below. And if you're looking for "top" venues or recommended venues, use the explore endpoint instead.

If lat and long is provided, each venue includes a distance. If authenticated, the method will return venue metadata related to you and your friends. If you do not authenticate, you will not get this data.

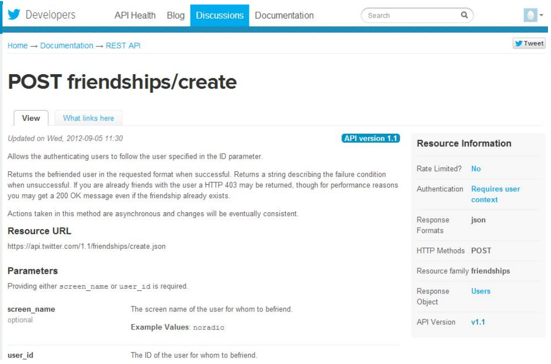
Note that most of the fields returned inside venue can be optional. The user may create a venue that has no address, city or state (the venue is created instead at the geolat/geolong specified). Your client should handle these conditions safely.

You'll also notice a stats block that reveals some count data about the venue. herenow shows the number of people currently there (this value can be 0).

This endpoint is part of the venues API.

HTTP Method	GET
Requires Acting User	No (learn more)

figure 1: Foursquare API - Search Venues



POST friendships/create

Updated on Wed, 2012-09-05 11:30

Allows the authenticating users to follow the user specified in the ID parameter.

Returns the befriended user in the requested format when successful. Returns a string describing the failure condition when unsuccessful. If you are already friends with the user a HTTP 403 may be returned, though for performance reasons you may get a 200 OK message even if the friendship already exists.

Actions taken in this method are asynchronous and changes will be eventually consistent.

Resource URL

https://api.twitter.com/1.1/friendships/create.json

Parameters

Providing either `screen_name` or `user_id` is required.

screen_name optional	The screen name of the user for whom to befriend. **Example Values**: `noradio`
user_id	The ID of the user for whom to befriend.

Resource Information

Rate Limited?	No
Authentication	Requires user context
Response Formats	json
HTTP Methods	POST
Resource family	friendships
Response Object	Users
API Version	v1.1

figure 2: Twitter API - friendships

Methods:

There are four python scripts to collect and manipulate data (See 3). After running the four scripts, a txt file will be created for generating a tag-cloud image on Tagul.com. Now I will go through the steps of the workflow.

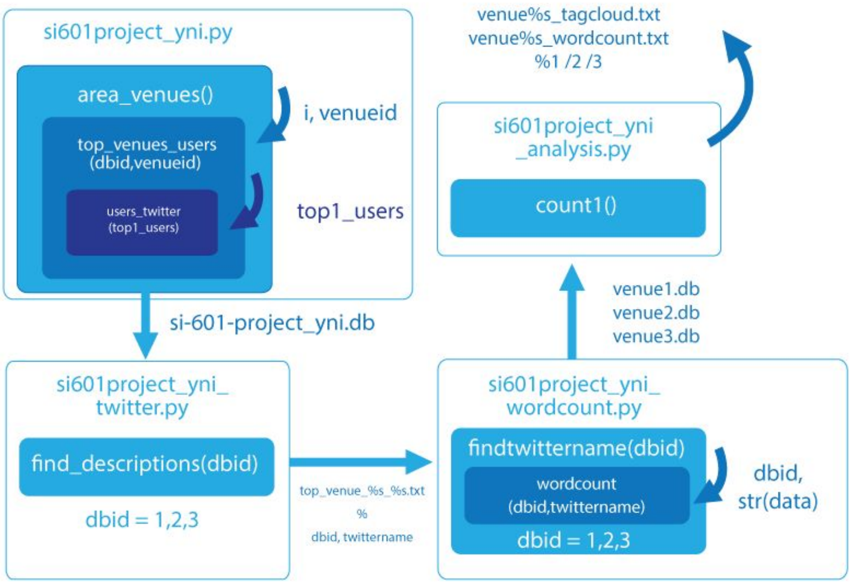
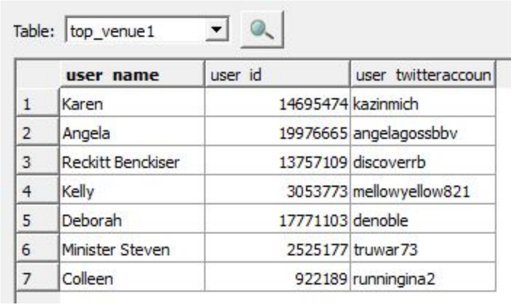


figure 3: Data collecting and manipulation workflow

Step1: Choose one category you want to search, and define the searching criteria (latitude and longitude, Intent: browse, radius, Category) on Foursquare API “Search Venues” page, and copy the url into the script. In the modified version, you can choose three different categories.

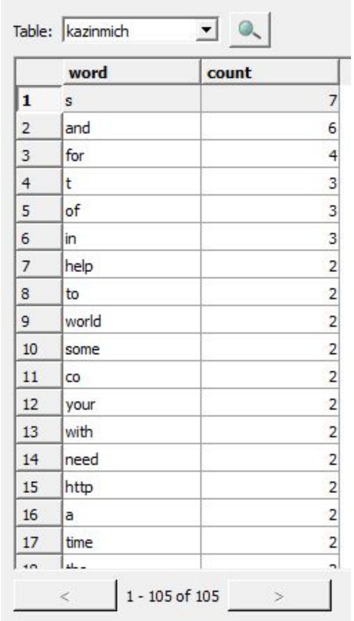
Step2: Run script “si601 project_yni.py” to find the top3 venues with checkin accounts in a certain category. In the modified version, the script will find the top1 venue in the three selected venue categories. The script will create a database called “si601-project_yni.db” with three tables for the three venus. For each venue, the script will find the mayor and at most 20 users who left a tip there(Foursquare API can only show at most 20 tips for a venue), and it will look for if the user has a twitter account. If they do, the script will write at most 10 users' data (Due to the rate-limit on Twitter) into that venue's table with users full name, their foursquare id and twitter account names (See figure 4.).



	user name	user id	user twitteraccoun
1	Karen	14695474	kazinmich
2	Angela	19976665	angelagossbbv
3	Reckitt Benckiser	13757109	discoverrb
4	Kelly	3053773	mellowyellow821
5	Deborah	17771103	denoble
6	Minister Steven	2525177	truwar73
7	Colleen	922189	runningina2

figure 4: si-601-project_yni.db -- table: top_venue1

Step3: Run the script "si601project_yni_twitter.py". The script will read the selected table in the database, pull out the users' twitter account name, get descriptions of the users and at most 10 of the friends they began following recently, and write them into a .txt file in the naming format of "top_venue_(venue number)_(twitter account).txt". If the twitter account is invalid, it will print out the error message and won't create the .txt file for the user.



	word	count
1	s	7
2	and	6
3	for	4
4	t	3
5	of	3
6	in	3
7	help	2
8	to	2
9	world	2
10	some	2
11	co	2
12	your	2
13	with	2
14	need	2
15	http	2
16	a	2
17	time	2

Step4: Run the script "si601project_yni_wordcount.py". The script will read the txt files of twitter users descriptions, count each word matching the regular expression "[\w]+", sort the words according to frequency, create a database for each venue (venue1/2/3.db), create tables named after the Twitter accounts, and write the word and count into the it (See figure 5.). In this step I faced a problem when I was using MRjob, since I have to read and write every txt file and I need to remove the """for each word. I finally solved the problem by replaying MRjob with another python script to automatically read every txt file created in prior steps and write the results into the database.

figure 5: Table for words and counts.

Step5: Run the script "si601project_yni_analysis.py". This script will read the venue database, create a list to put in all the "unique word" of each description, filter out meaningless words, and count the number of rest of the words in each venue. The reason why I didn't count each word directly from all the descriptions is to avoid bias caused by a particular user. For example, if the word "football" appears 15 times in the descriptions collected for a twitter user, it may lead to misinterpretation that the 10 users for a venue have a common interest of football. So only counting unique word in descriptions for a user and see the overlap for the 10 users can better reflect if their interests overlap in some ways. The script “si601project_yni_analysis.py” will generate two txt files for each venue, one is for creating a tagcloud on Tagul, with words repeated according to their frequency in the analysis; another one is for plain reading, with word and the number of its frequency.

Results:

After I finished the first version of the project, I analyzed the food place in Ann Arbor (Latitude and longitude: 42.281262,-83.740049; Intent: browse; radius:5000; Category: Food) with top3 checkins. After getting Twitter accounts of 10 users who left tips on foursquare for these restaurants, I count the word in the descriptions of them and 10 of their friends. The top five words for Zingerman are “more-4; love-4; life-4; ceo-4, account-3” (See figure6). The top five words for Grizzly Peak Brewing Co. are “michigan-6; news-6; tweets-6; like-6; food-5”(See figure7). The top five words for Conor O’Neill’s are “world-6; new-5; more-5; university-5; detroit-5” (See figure8). From the results I assume Zingerman’s customers are more interested in life and art, and Grizzly’s customers are more interested in news and Michigan, and Conor’s customers are more concerned about business of world and detroit. However, due to the small amount of data, it is hard to approve my assumption.

In order to see if the analysis can represent the whole customer population in some degree, I modified the original scripts a little to make them be able to check venues of different categories. I changed the Foursquare Search Venue variable “CategoryId” and tried to find the restaurant with No.1 checkins of Japanese restaurants, Mexico restaurants and German restaurants (See figure9, 10, 11). However, in the results I didn’t see a great pattern in the customers and their interests on Twitter. I tried to search for top 1 venues in the categories of “Engineering Buildings”, “Art Buildings” and “Math Buildings”, but since these venues have too few tips (less than 5 tips), I failed to get enough twitter user accounts to make the result more representative. In short, I came up with the conclusion that due to the limitation of Foursquare API and Twitter API., the scripts now cannot find special relationship between the places and the people's interests on Twitter. However, if I can get a whole list of users who checked in one place and can be pull out a large-enough data from Twitter, the results can be more representative and meaningful.

Although the limitations of APIs make it hard for me approve the meaningfulness of the results, I have another interesting finding: Analyzing the descriptions of twitter users and their friends (people they are following) will get better results on users’ interests than directly analyzing users’ tweets. Once I was not sure if I should use users’ tweets or use descriptions of users and their friends to dig into their interests and characteristics. So I tried both of two to see which one works better, and the results proved that the descriptions are more effective to reflect twitter users’ interests. For example, for the Twitter user “Ghostly”(See figure 12), you can see the analysis of descriptions highlight the word “Music” while the analysis on tweets is meaningless. Actually “Ghostly” is a recording company. I think the reason why tweets cannot reflect twitter user better is that the tweets are usually about slices of daily life, and a lot of meaningful information is hidden inside links.

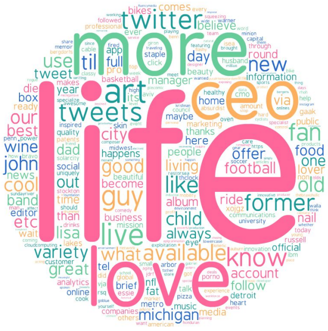


figure 6: No.1:
Zingerman's Delicatessen

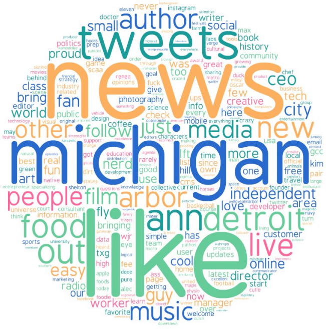


figure 7: No.2
Grizzly Peak Brewing Co.

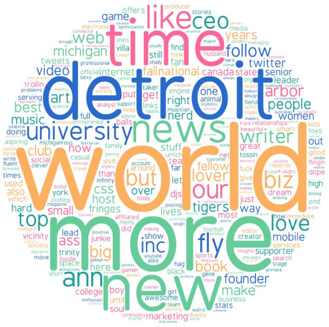


figure 8: No.3
Conor O'Neill's.

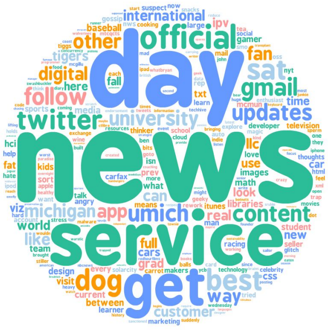


figure 9: No.1 Japenese
Restaurant: Sadako

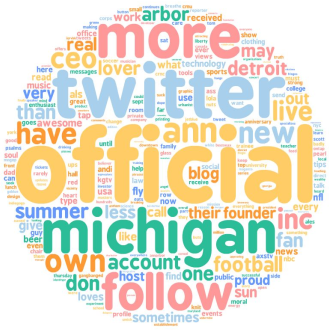


figure 10: No.1 Mexico
Restaurant: Chipotle
Mexican Grill

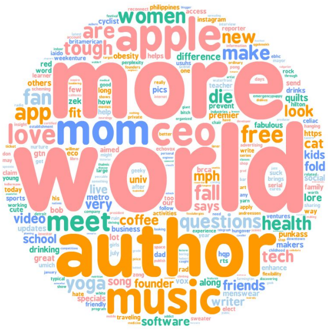


figure 11: No.1
Germany Restaurant:
Heidelberg

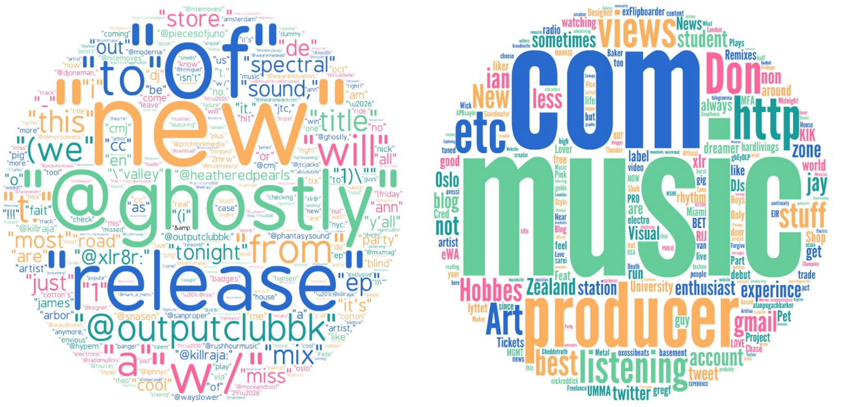


Figure 12: Difference of the analysis between tweets and descriptions of user and their friends (Analysis of tweets in the left and descriptions of the user and friends on the right).