SMU
SINGAPORE MANAGEMENT UNIVERSITY

Singapore Management University

Master of IT and Business (Analytics)

Module: ISSS606-Social Analytics and Applications-G1

Social Analytics Project:
A Graph Based Recommender System

Submitted by: Group 5

Chris Thng, Ong Han Ying, Tham Jun Quan,

Wesley Chan Lun, Yang Cheng, Ye Zhibo

Lecturer: Professor Dai Bing Tian

Submission Date: 30/07/2017

1. Introduction

This project aims to develop a restaurant recommendation system by analysing the behaviour and relationships such as relationships between users, types of cuisines they enjoy, sentiments towards restaurant(s) and areas they frequent. This will be conducted based off the historical data taken from Foursquare.

We will be using various Social Network Analysis (SNA) techniques such as clustering methods to detect communities within this dataset. These clusters will then form the basis of our approach in designing our recommendation system. With the results generated from the clusters, we utilized a centrality measure technique proposed by Opsahl et al. (2010) to develop the recommendation system. The system was then evaluated based on a Mean Reciprocal Rank approach.

2. Motivation

Our inspiration comes from a common problem many people in Singapore face – where and what to eat for lunch. Singaporeans generally love their food and look forward to a delicious plate of their favourite food whilst in the company of their friends or colleagues. However, after a period of time most people tend to eat the same foods or frequent the same few restaurants or eateries.

In Singapore, we are famous for the great variety and quality of food, but why are locals not being more adventurous to explore and find new types of food to enjoy? This is mainly due to the hectic lifestyle many locals lead, they are either bogged down by work or family and do not have much time to think about the next place to have their meal.

This is where Jiak*Bot 2.0 comes in. It aims to suggests places to eat through the chatbot to provide an interactive user experience and to enable the user to make a more informed choice. It is built using Natural Language Processing (NLP) together with Social Network Analysis (SNA) techniques.

3. The Dataset

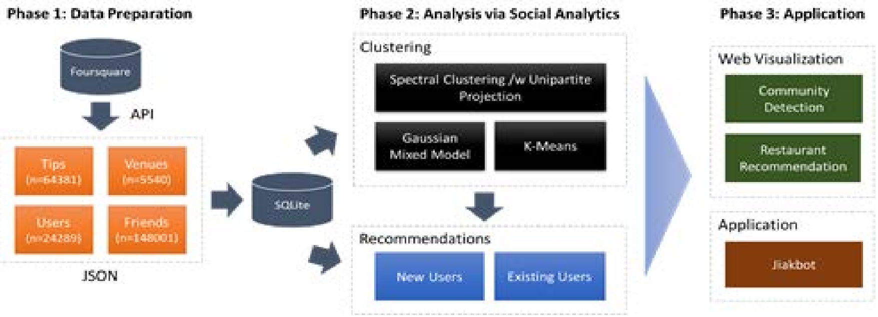
We have managed to collect data from Foursquare of over 5,540 Singaporean businesses. The major characteristics of this dataset is as follow as:

1. 64,381 Tips – the number of total comments collected for all 5,540 venues
2. 5,540 Venues – the number of business/restaurants
3. 24,289 Users – the number of people using Foursquare in Singapore
4. 148,001 Friends – the number of friends all users have

The data was stored in JSON format and stored in SQLite as the storage database. SQLite is able to handle the amount of data well and is easy to use, which was appropriate to use for this project.

4. Methodology

To meet the objective of the analytic work, we designed and implement the following workflow;



In all, we proposed a 3 phases workflow such that:

Phase 1: We will identify our data source, and prepare our data ready for further analysis

Phase 2: Through 3 relevant factors that influence decision in dining options, we conduct 3 different clustering for each of these factors to identify distinct community and/or behaviors in each of these clusters. With the results from the clustering, we then proceed to make use of the obtained results from the analysis to design a recommender system for both new and existing users

Phase 3: With the results from both clustering and recommendation, we develop an online web dashboard via D3.js and Tableau to put these results into use for the businesses, and lastly a chatbot, to display the results and recommendations to end-users in an interactive manner

Source control was done using GitHub with the repository available at: https://www.github.com/junquant/saproject

4.1. Data Preparation & Flow

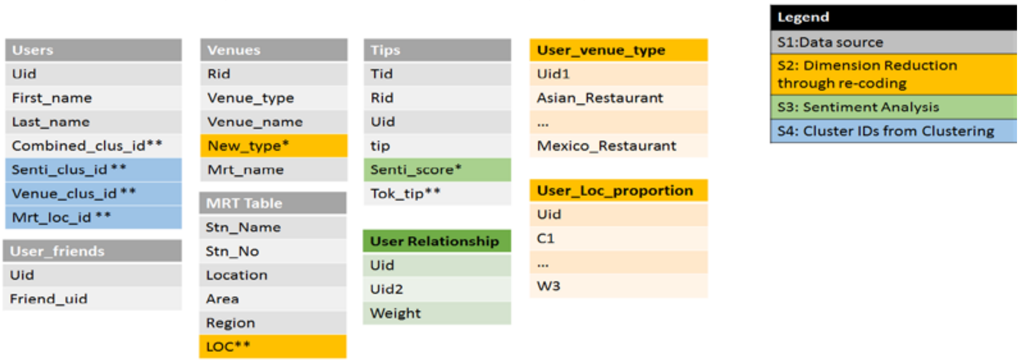
Data Sources: In all, the key data sources that we have used for this analysis, as below;

1. ***Four Square API*[1]*:*** Through Foursquare's 3 main core APIs of "Tips", "Venue", "Users", we pull out the data in JSON format and store them into 4 main tables of "Tips", "Venue", "Users", and "Friends" into a database, SQLite.

2. ***data.gov.sg*[2]*& Wikipedia*[3]*:*** Through both websites, we identified the towns in Singapore, and compare it against the region that the town belong to – for geographical clustering.

3. ***mytransport.sg*[4]** through the API, we identified the latitude and longitude of the MRT stations in Singapore, and used them as different initiation points to crawl data in Four Square API.

Data Flow: In all, the database is updated based on the 4 key steps as below;



Users
Uid
First_name
Last_name
Combined_clus_id**
Senti_clus_id **
Venue_clus_id **
Mrt_loc_id **

User_friends
Uid
Friend_uid

Venues
Rid
Venue_type
Venue_name
New_type*
Mrt_name

MRT Table
Stn_Name
Stn_No
Location
Area
Region
LOC**

Tips
Tid
Rid
Uid
tip
Senti_score*
Tok_tip**

User Relationship
Uid
Uid2
Weight

User_venue_type
Uid1
Asian_Restaurant
...
Mexico_Restaurant

User_Loc_proportion
Uid
C1
...
W3

Legend
S1:Data source
S2: Dimension Reduction through re-coding
S3: Sentiment Analysis
S4: Cluster IDs from Clustering

Step 1: Database is updated with data from the data sources directly.

Step 2: Location (of Mrt stations) and Restaurant Types have too many categories, and therefore; they are being re-coded manually to reduce the total no. of categories.

Step 3: With the data, a sentiment analysis is conducted and the database is updated with the score, and with further analytic work, a weight is assigned. ***(refers to section 5.2 for more details)***

Step 4: with the data from the above, clusterings are conducted and the database is updated with the final cluster IDs. ***(refers to section 5.2.2 for more details)***

The output from this database is then being used to develop applications, for visualization of results and development of chatbot.

[1] https://developer.foursquare.com/docs/

[2] https://data.gov.sg/dataset/resale-flat-prices

[3] https://en.wikipedia.org/wiki/Regions_of_Singapore

[4] https://www.mytransport.sg/content/mytransport/home/dataMall.html

4.2. Community Detection in Foursquare Network

Considering our goal of providing venue recommendations for each user, in this section we aim to find communities in the Foursquare network (i) who have similar taste, (ii) who usually go to venues located in certain geographical areas, and (iii) who have similar sentiment score for a given set of venues.

There is no universally accepted definition of community. The definition often depends on the specific system at hand and/or application one has in mind. A very comprehensive study on this topic was performed by Fortunato [1].

Nevertheless, a general intuition is that a community is constituted by a group of cohesive nodes, in the sense that there is much more edges / links "inside" the community than edges linking nodes of the community with the rest of the graph.

The literature offers a wide range of algorithms and methods that can be applied in order to identify communities in network structures, such as K-Means clustering, Hierarchical clustering, Spectral clustering, Latent Space models, and Modularity Maximization.

4.2.1. Clustering based on Venue Location

In this section, we seek to identify communities of Foursquare users who usually go to venues that are geographically located near to each other. This is based on the notion that people tend to eat at places located closer to them, be it where they live, where they go to work or go to school, rather than travel a farther distance. Insight from the geographical cluster is thus a vital part of understanding Foursquare user's behaviour and will be use to provide recommendations for new venues.

We set the users as a set of nodes, the geographical area as another set of nodes, and the visits as the edges linking from the users to the venues. Each of the venue is mapped to a set of 20 geographical area in Singapore based on their geographical coordinates (latitude and longitude), while the edge weight is modelled as the proportion of a user's overall visits to a geographical area – as proxied by the number of comments / tips a user wrote for a venue, since the actual number of visits to the venue is not directly captured in Foursquare. One limitation here is that users may not write tips for all the venues they go to. We, thus, constructed a directed bipartite graph connecting the users to the venues. We next use K-Means to group similar users according to where they usually go to eat, based on the set of 20 geographical areas as our feature vector.

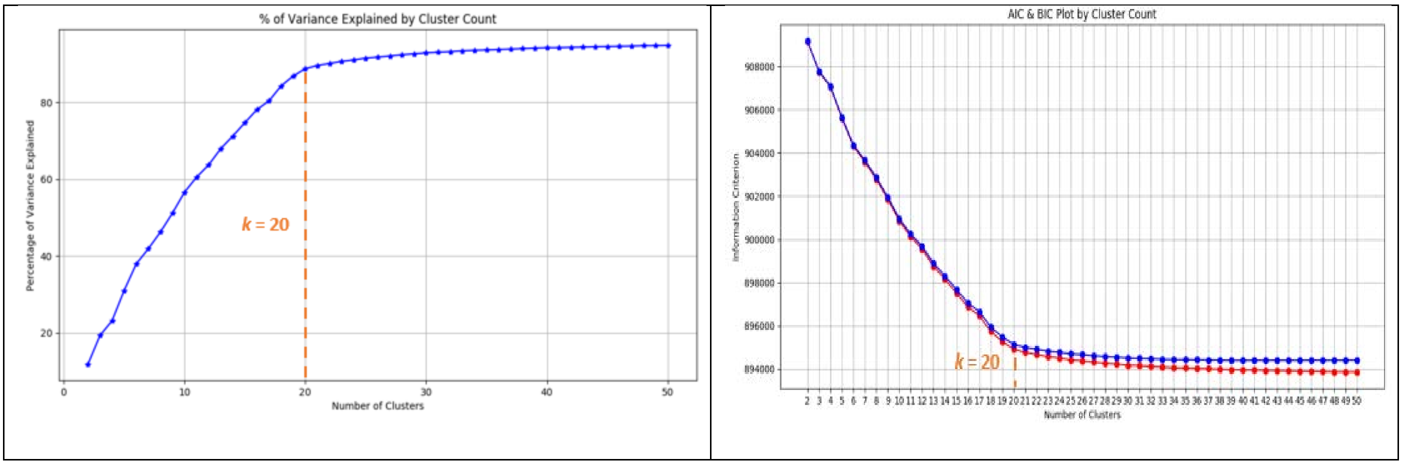
K-Means is the most known and studied method for clustering analysis. K-Means clustering implements least-square partitioning in classifying an input data set into an a priori fixed number, *k*, of groups. The goal is to separate points into *k* clusters such as to minimize the total intra-cluster distance, or squared error function:

$$\sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - c_i \right\|^2$$

where, indicates the subset of points of the *i*-th cluster and its centroid. The K-Means algorithm starts by initializing *k* cluster centres. It then iterates between an assign step, where each sample is assigned to its nearest centroid, and an update step, where each centroid is updated to become the mean value of all the samples that are assigned to it. This iteration continues until the cluster assignments no longer change.

As the ground truth labels are unknown, there are various options for selecting the number of clusters *k* to use. We explored two approaches: (1) the percentage of variance explained method and (2) the information criterion approach using the Akaike information criterion (AIC) and the Bayesian information criterion (BIC).

As illustrated in Figure XX and XX, both approaches indicate that the optimal *k* is 20. The location of a bend (knee) in the percentage of variance approach plot is generally considered as an indicator of the appropriate number of clusters. While the optimal clustering using the information criterion approach is based on the one with the lowest value of AIC and/or BIC. The clusters formed with *k* = 20, are quite distinct.



4.2.2. Clustering based on Venue Type

Clustering based on venue location is very intuitive because users tend to eat near where they live, where they go to work or go to school etc. But location is not the only factor that describe users dining behaviour. In order to improve our venue recommendations, we also consider the type of venues the user like.

Similar to the analysis in section 5.2.1, we set the users as a set of nodes, the venue / cuisine type as another set of nodes, and the visits as the edges linking from the users to the type of cuisine. Each of the venue is mapped to a set of 20 distinct venue types, down from the original one hundred types classification on Foursquare. The edge weight is modelled as the number of times a user visited each type of venue. We next use K-Means to group users with similar taste based on the type of venue they go to as our feature vector.

Following section 5.2.1, we set the cluster size, k, to range from 2 to 30, and use the AIC and BIC criterion to select the optimal *k*. The AIC/BIC plots (not shown due to space constraints) shows that the optimal K is 9.

4.2.3. Clustering based User Relationship projected from Sentiment Scores

We next try to identify communities of Foursquare users who have similar sentiment score for a set of venues using a similarity matrix. The similarity matrix is constructed by projecting the bipartite graph between users on the one hand represented as a set of nodes, the venues as another set of nodes and the sentiment score as the edges linking from the users to the venues, into a unipartite graph. An individual's sentiment score for a given venue is derived from the compound score on the individual's comments (defined as 'tips' in Foursquare) on the venue. This is obtained using the Sentiment Intensity Analyzer module in NLTK as we wish to distinguish between good and bad reviews, and accordingly assign the sentiment score as the edge weight connecting a user to a venue.

To project the bipartite graph into a unipartite graph, we defined a weight function that take into account not only the number of common restaurants / venue shared between each pair of Foursquare users, but also their sentiment score / rating towards the same venue. Thus, individuals who go to the same venue and have similar sentiment score towards the venue would be more similar than individuals who have rated the venue differently, and the edge between two users in the resulting unipartite graph means that these two users went to common venues.

Spectral clustering is used for this analysis because it first maps the original data points into a lower-dimensional space, obtained from the eigenvectors of the graph Laplacian matrix. The change in the representation of the data enhances the cluster properties and improves cluster detection. After embedding the original data into a lower-dimensional representation, the final clustering result is obtained by running a Gaussian Mixture Model ("GMM") on that representation. This technique is similar in spirit to k-means clustering. This method accounts for the fact that a point may belong to two or more clusters at the same time.

The cluster size, k, ranging from 2 to 30, is performed on the user similarity matrix projected from the sentiment score. Similar to section 5.2.1, the AIC and BIC criterion are used as an internal clustering validation measure to select the ideal *k*. The AIC/BIC plots (not shown due to space constraints) shows that the optimal K is 6.

4.3. Designing & Developing a Recommendation Framework

2 bipartite graphs were built and evaluated against each other for providing recommendations. The first graph (full history graph) is built using the full history of venues visited by the users with edges weighted by the sentiment scores each user gave to the venues they visited. The second graph (recent history graph) was built by using the recent 6 visits by each of the users iteratively. At each iteration, the latest venue was removed and the most recent venue not included in the 6 venues was included. This was done until there are no earlier venues to be appended to the 6.

Each of the user nodes was then assigned a cluster based on the earlier clustering results. For each cluster, the venue nodes were assigned a centrality score using a centrality measure proposed by Opsahl et al. (2010) [2]. The centrality measure is defined by:

$$C_D^{w\alpha}(i) = k_i \times \left(\frac{s_i}{k_i}\right)^{\alpha} = k_i^{(1-\alpha)} \times s_i^{\alpha}$$

where C is the centrality score, k is the number of edges connecting to the node and s is the sentiment score of the edge.

3 sets of recommendations based on the user history, clusters and centrality measure of the venues were proposed for each user. Each set of recommendation uses a different combination of the detected clusters or a different graph for comparison purposes. For each user, the latest venue visited is being regarded as the relevant choice and the recommendations were then evaluated by calculating the Mean Reciprocal Rank (MRR) of the relevant recommendation. MRR is defined as:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

where Q is the number of users.

The results of the recommendation and evaluation is shown below.

Graph Used	Clusters Used	MRR	Description
Recent History	Venue	0.0369	Relevant recommendation ranked approximately #28
Full History	Venue	0.0519	Relevant recommendation ranked approximately #20
Full History	Venue + Rating + Location	0.0672	Relevant recommendation ranked approximately #15

From the above results, we see that in general, the more information about a user available to make recommendations, the better is the recommendation.

5. Methodology

Putting the analysis together, the output of the results is put into visualization tools such as D3.js (open source javascript library for visualization), and tableau for further interpretation of the results. This series of visualization also have the following objectives for the business users, as below;

1. **Nature of relationships in graph:** Understand the nature of the relationship between the users and the restaurants (the nature of the dataset) via network graph (by d3.js)

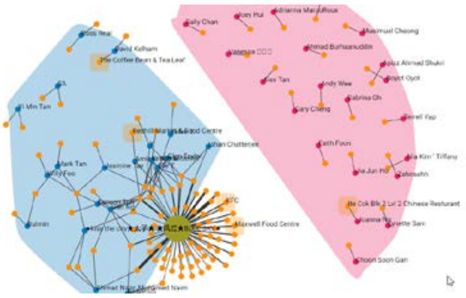
2. **Interpretation of clustering results:** Distinct community represented by each cluster via interactive bipartite graphs (by D3.js), and geographical map from tableau.

3. **Visualize the graphs used to develop recommendation frameworks:** interactive treemap and network graph are developed (by D3.js) to help the business users to visualize the impact of (a) combined IDs on existing users, and (b) impact of the network of friends of users on recommendation.

All visualization work and results are hosted at https://junquant.github.io/saproject/dashboard.html

5.1. Nature of Relationships in Graph

In all, 2 main types of users-venue relationship are identified as below;



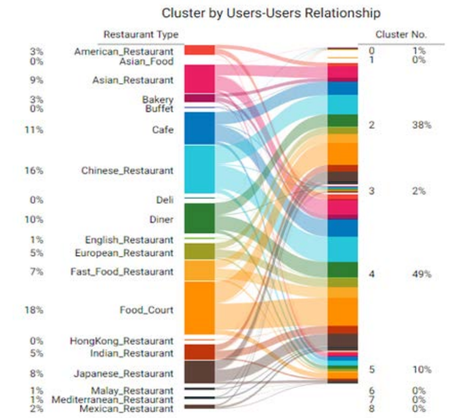
1. **Unipartite relationship:** each user is connected to 1 unique restaurant name only. (Pink)
2. **Bipartite relationship**: each user and restaurant is connected to more than 1 user and/or restaurant (Blue)

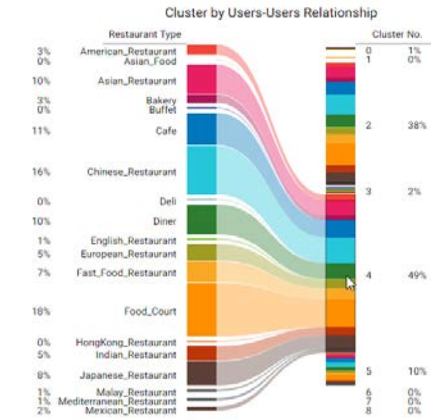
Implication: Understanding this nature of the relationship in the graphs will then help us to understand the results of the clusters. As a result, users with unipartite relationship are manually grouped into 1 cluster, since it is not possible to be clustered under user-user relationship clustering.

5.2. Interpretation of Clustering Results

In all, 3 different graphs are developed to help to interpret the results from the clustering.

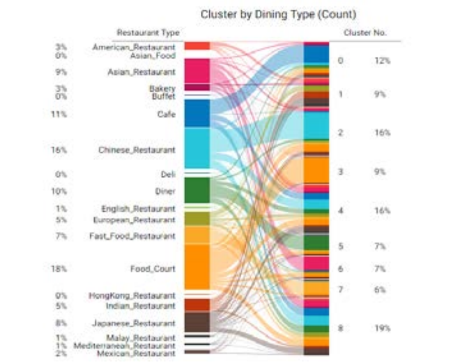
(1) Bipartite graph for clustering for user-user relationship: we can see from the graph below that;

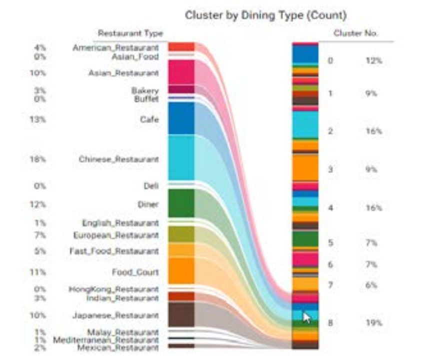




- A total of 6 clusters are identified under this clustering.
- Cluster 2 and 4 have the largest member by % of 49% and 38% respectively.
- Focusing into cluster 4, the proportion of the restaurant types are distributed across all types, with highest proportion of 18% found in Food Court.

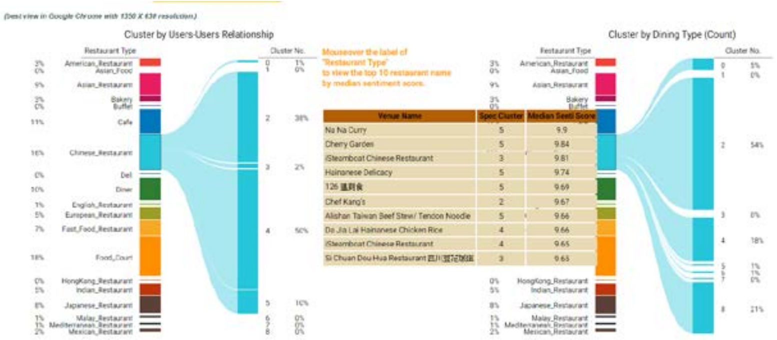
(2) Bipartite graph for clustering by Dining Type: we can see from the graph below that;





- A total of 9 clusters are identified
- Cluster 8 have the largest member by 19%
- Focusing into cluster 8, the proportion of the restaurant types are distributed across all types, with highest proportion of 18% found in Chinese Restaurant.

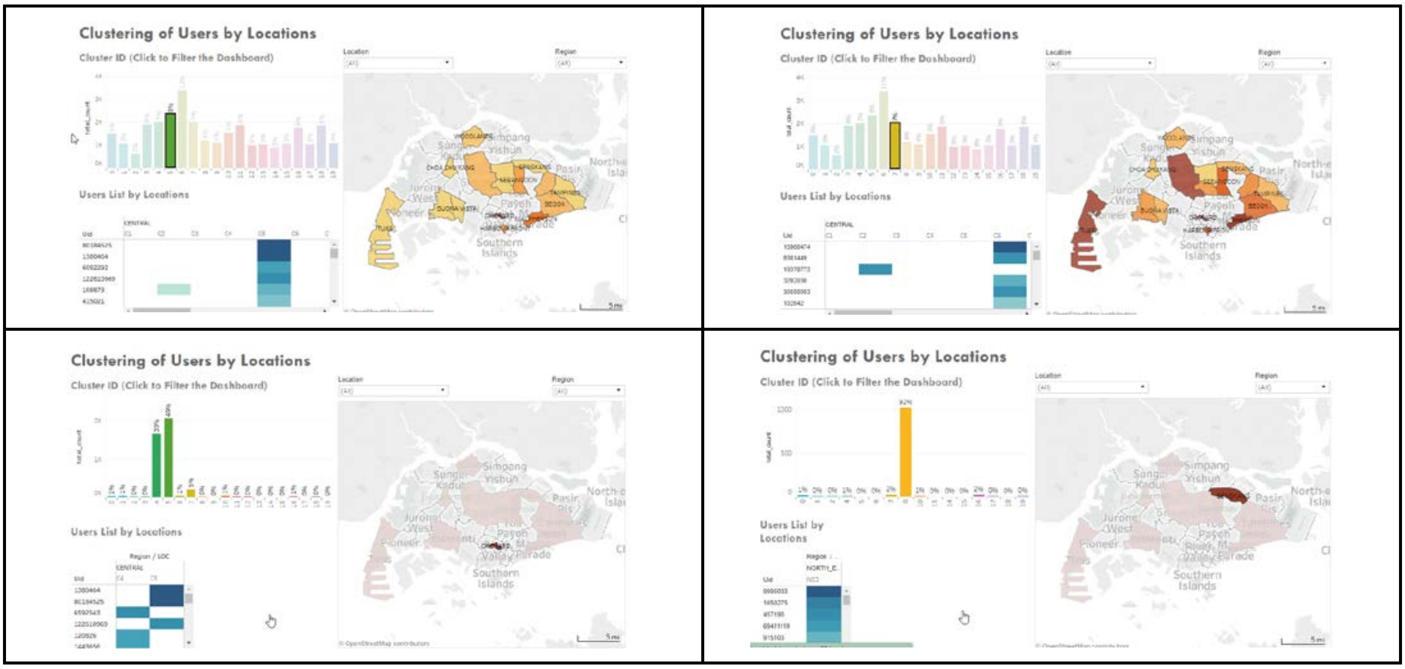
Putting both analysis together: for each of the restaurant type, we can identify its distribution in both type of clustering. For example, for the graph below; we see that for Chinese Restaurant, it is concentrated in cluster 2 and 4 from user-user relationship clustering, and cluster 2 in clustering by Restaurant Type. Also, we can see the top 10 restaurant names with highest median sentiment score across different clusters.



Venue Name	Spec Cluster	Median Senti Score
Na Na Curry	5	9.9
Cherry Garden	5	9.84
iSteamboat Chinese Restaurant	3	9.81
Hainanese Delicacy	5	9.74
126 溫到食	5	9.69
Chef Kang's	2	9.67
Alishan Taiwan Beef Stew/ Tendon Noodle	5	9.66
Da Jia Lai Hainanese Chicken Rice	4	9.66
iSteamboat Chinese Restaurant	4	9.65
Si Chuan Dou Hua Restaurant 四川豆花饭庄	3	9.65

Implications: With the above analysis, these can help food businesses (by type) to identify potential customers via the members in these clusters, and also; to identify the food preference of the members from each of the cluster.

(3) Geographical graph for clustering by Location: we can see the distribution of the 19 clusters of users by location. For instance, referring to the graph below; we can see that (1) While cluster 5 (8% of total) have concentration of users who prefer to eat in orchard, but this distribution differs from cluster 7, with concentration in Chua Chu Kang and Tuas. (2) Users who prefer eating in orchard is concentrated in cluster 4 and 5, while users who eat at Seng Kang are concentrated in cluster 8.


Clustering of Users by Locations
Cluster ID (Click to Filter the Dashboard)
Location
Region
total_count
Users List by Locations
Uid
CENTRAL
Clustering of Users by Locations
Cluster ID (Click to Filter the Dashboard)
Users List by Locations
Clustering of Users by Locations
Cluster ID (Click to Filter the Dashboard)
Users List by Locations
Region / LOC
Clustering of Users by Locations
Cluster ID (Click to Filter the Dashboard)
Users List by Locations
NORTH_E
Sungei Kadut
Simpang
Yishun
Pasir Ris
Jurong West
Pioneer
Southern Islands
Paya Lebar
Marine Parade
5 mi

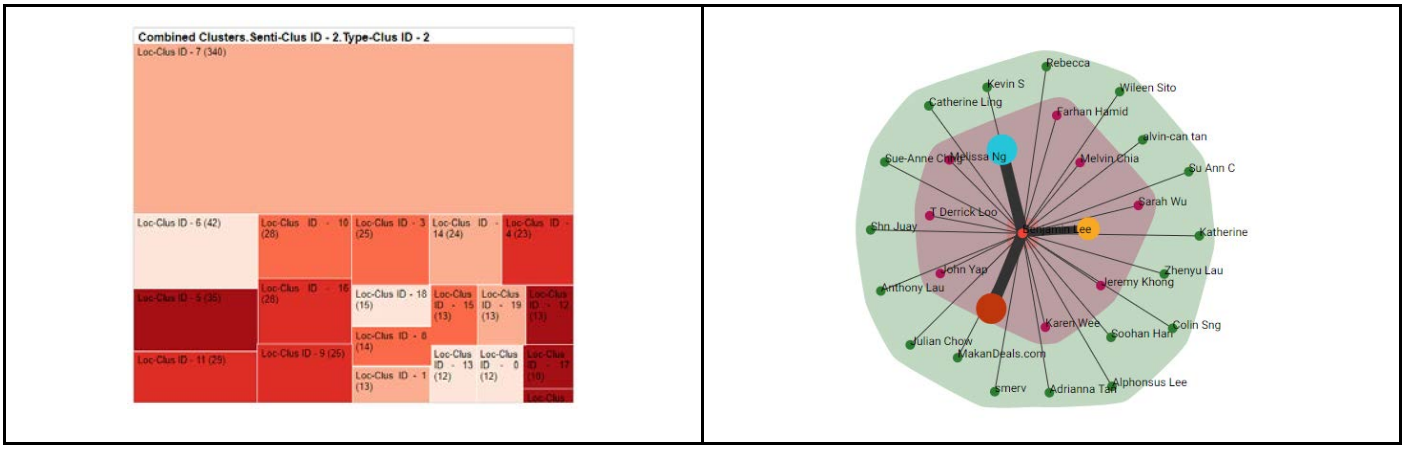
Implications: With the above analysis, it helps to identify distinct community of eating locations of users and these analysis can help a business owner to determine the cluster of users to target for marketing if a stall is set up in a specified location/ region.

(4) Graphs developed for recommendation framework: through visualization, we are able to understand the graphs that help to develop recommendation for each user, as below;

1. Existing Users: via the tree map, we can see the size of each of the combined IDs and this will also refer to the no. of similar users that the recommendation is being made reference with.

2. New Users: Since depending on the combined cluster ID of a new user may not be precise due to small dataset even from the user him/herself, we will refer to the biggest cluster_combined_id that his/her friends belong to.

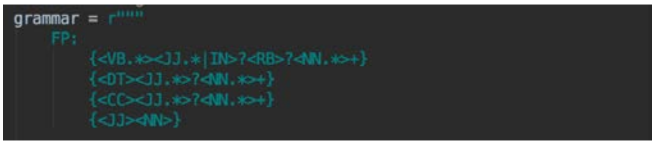
From the network graph below, it makes it easier for us to visualize that for this user, he/she has more friends from the green cluster than the other cluster. Therefore, recommendation will be made via referring to this cluster. A similar network graph for each of the new users are implemented for each of them.



Next, these graphs are also being applied to be referenced with to develop a user application for food recommendation for each of the users.

5.3. Development of User Application – JiakBot

JiakBot is a chatbot developed using natural language processing techniques and provides graph based recommendations based on an user's history. It aims to demonstrate that the analysis performed can be applied in a practical scenario. Natural language processing tasks include tokenizing and tagging the part of speech for each statement entered by the user. A Python dictionary was then used to store the tokens, cleansed text, verbs, nouns, adverbs, adjectives and pronouns. A Random Forest classifier was also used to determine if the input is a statement or a question. Next, Information Extraction was performed and a grammar syntax was then created to identify if certain phrases is a "Food Phrase". The grammar used is shown below. Extracted food phrases are then looked up a list of known foods extracted from Foursquare to determine if there is such a food in the crawled database.

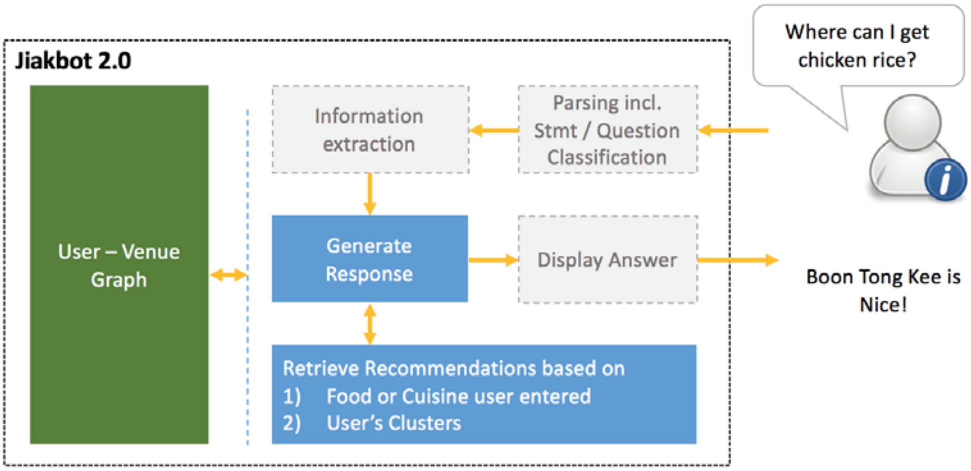


```
grammar = r"""
    FP:
        {<VB.*><JJ.*|IN>?<RB>?<NN.*>+}
        {<DT><JJ.*>?<NN.*>+}
        {<CC><JJ.*>?<NN.*>+}
        {<JJ><NN>}
```

After understanding the user's input and the type of food an user would like. The next step would be to generate a response for the user. For generating the response, we made use of the bipartite graph created during our analysis and developed 2 functions for recommendations.

1. get_venue_by_rids - this function returns a restaurant from the list the venues that serves the food that was requested by the user. The returned restaurant is the top ranked restaurant in terms of centrality in the cluster that the user belongs to.
2. get_venue_by_uid - this function attempts to suggest a restaurant based on the user's cluster if no food was entered. Again, the top ranked restaurant within the cluster in terms of centrality will be returned.

The parsing, information extraction, and recommendation process is summarized by the diagram below.


Jiakbot 2.0
User – Venue Graph
Information extraction
Parsing incl. Stmt / Question Classification
Generate Response
Display Answer
Retrieve Recommendations based on
1) Food or Cuisine user entered
2) User's Clusters
Where can I get chicken rice?
Boon Tong Kee is Nice!

The bot is then hosted on Telegram (https://web.telegram.org) and exposed publicly with the name @jiak_bot.

6. Discussion – Challenges Faced

Firstly, during the collection of data, we realized that Foursquare did not provide check-in data. And this data was fundamental for one of our clusters to perform correctly. Hence, we used the number of comments or reviews they left for the venue as a proxy to estimate the number of times they went to a certain place.

Secondly, we realized that some of our attributes had dimensions that were too big such as location (e.g. up to 100 areas, Jurong East, Jurong West, Jurong North, Jurong South). This would have an effect on our clusters in terms of analysing as well as visualizing. We decided to manually reduce the dimension of these attributes by using, for example the Urban Redevelopment Authority (URA) urban planning map to group certain locations together (e.g. Raffles City, City Hall, Tanjong Pagar into Central + CBD).

Thirdly, we initially set a random K for our clustering. However, the results were not interpretable and hence, we used the AIC/BIC method to get the optimal K for our clusters. This vastly improved the interpretability of the results we obtained.

Fourthly, we used multiple cluster techniques for each of three clusters. After conducting our own research based on the appropriateness of each cluster technique on certain types of data, its dimensions and the objective, we were able to pick the most appropriate one for each cluster.

Lastly, with a recommendation system, there should be an evaluation of the accuracy of our results. However, being fairly new to this area of developing recommendation systems, the team deliberated over different evaluation methods before deciding on using the Mean Reciprocal Rank technique.

7. Conclusion and Future Works

Building a recommendation system using both NLP and SNA techniques was an interesting and challenging task by having to design the methodology, put them together and work with each other. During the process of our research and development, we found numerous research papers for chatbot source codes as well as recommendation systems that were shared online and given more time, one should leverage on existing works and findings. However, as each language and country has its unique nuances developing a localised chatbot requires customisation. Additionally, with a different set of objectives from other recommendation systems, we could borrow certain ideas but ultimately develop our own methodology in solving our objective.

In future, Jiak*Bot 2.0 could be further trained to develop itself further in an unsupervised environment as presently, many of its functions are utilizing rule-based functions to run certain aspects of it such as its internal state machine and responses. During this developmental period, the number of inputs and trials we could test on this bot was limited and hence was not appropriate to try and train the bot with the inadequate amount of data input.

In summary, we believe that we can expand Jiak*Bot 2.0 globally to cover other countries such as Malaysia, Indonesia and even the United States. The techniques and models used in this version can be applied to the dataset of other countries, and with a few minor changes to aspects such as the code in terms of clustering and dimension reduction, we foresee Jiak*Bot being able to provide robust recommendations to the international market too.

The potential for this Bot is great, the business opportunities available to it include and not limited to:

- Subscription-based Telegram bot (either paid app or in-app advertisements)
- Restaurants that are looking for a place to set up shop in (e.g. can identify the target audience)
- Government planning (e.g. effects of redevelopment/planning of amenities)
- Paid application (e.g. listed on AppStore/GooglePlay)
- Incorporated into a messaging system (e.g. SMS for a recommendation, based on user history/others)
- Or even free to the public as a social service

8. Reefrences

[1] Fortunato, Santo. "Community Detection in Graphs", Physics Reports, 486 (2010), 75-174.

[2] Opsahl, Agneessens, & Skvoretz. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. Social Networks, 32(3), 245-251.