Graph Visualization and PageRank of the Yelp Social Recommender Network

Nond Hasbamrer
Department of Electrical Engineering
Columbia University
nh2518@columbia.edu

Abstract — **Yelp users traditionally interact with business listings by looking at star ratings and reading other user's reviews. However, manually scanning through pages of ratings and reviews isn't scalable. A city like Pittsburgh can contain over thousands of business listings and tens of thousands of reviews. I propose a network graph visualization of the Yelp social recommender network that uses PageRank relative node size to illustrate business importance and influence. To demonstrate the benefits of graph visualization, I created sample graphs using data from the Yelp Academic Dataset Round 8.**

Keywords- information visualization; visual analysis; network graph; recommender network; recommender systems;

I. INTRODUCTION

Yelp connects people with businesses by providing a way for users to communicate their experiences. Yelp users or "Yelpers" have contributed more than 115 million reviews for a variety of local businesses according to Q3 2016 investor data. On average, over 174 million unique visitors access Yelp's data each month [1]. Individual businesses can receive over thousands of reviews; Mon Ami Gabi, a French restaurant in Las Vegas, currently has over 6,300 reviews. As the Yelp community continues to grow, data volume and understanding poses an increasingly difficult challenge. A user can manually scan reviews for a restaurant with a couple hundred reviews, but manual scanning doesn't scale to thousands of reviews across multiple restaurants. Likewise, businesses cannot rely on manual review scanning to identify competition in dense markets. Even looking at a list of businesses with only the associated star ratings can be overwhelming when users have to flip through multiple pages of business listings.

To address both issues of data volume and understanding, I propose a methodology that utilizes IBM System G Graph Tools to present users with a network graph visualization of the Yelp Academic Dataset Round 8 [2]. The graph shows an indication of business star rating and popularity as well as how the network of businesses and reviewers are connected.

II. RELATED WORKS

Previous work on the Yelp Academic Dataset focused primarily on review text analysis and star ratings. Work by Huang, Rogers, and Joo [3] used an online Latent Dirichlet Allocation (LDA) algorithm to extract subtopics from user review text. The extracted subtopics could provide businesses with insights into what their customers cared about. Along the same trend, McAuley and Leskovec [4] combined topics generated from LDA with rating dimensions from latent-factor recommender systems. Their work provided a way to identify genres for products through the review text and identify reviews that users are likely to find useful. Linshi [5] incorporated star ratings into a modified LDA that identifies positive and negative adjectives associated with traditional topics such as "good service" instead of just "service". Gupta and Singh [6] performed collective factorization to determine review words related to categories and business attributes and used word cloud visualization for qualitative evaluation. Wang, Zhao, Guo, and North [7] presented a clustered layout word cloud as an alternative to manual review scanning. This technique is effective at summarizing individual businesses and enabling users to make quicker decisions about each business.

Even though these works pay meticulous attention to the text of the reviews, the network of businesses and users as a whole wasn't taken into account. Works cited by Kempe, Kleinberg, and Tardos [9] have shown that targeting a subset of influential individuals within a social network can trigger a large cascade of opinion adoption. In context of the Yelp social network, businesses can better utilize limited marketing resources by targeting a subset of influential consumers.

Wong, Liu, and Chiang [10] studied the efficiency of the Yelp social network based on two metrics: an individual's user experience and the network's efficiency at information propagation. Their work concluded that Yelp's network is quite efficient at disseminating information.

The combination of product and rating integration into the social network, network volume and efficiency, and public availability makes Yelp a great platform for recommender systems research. My contribution towards this area of research is a methodology for visualizing Yelp's social

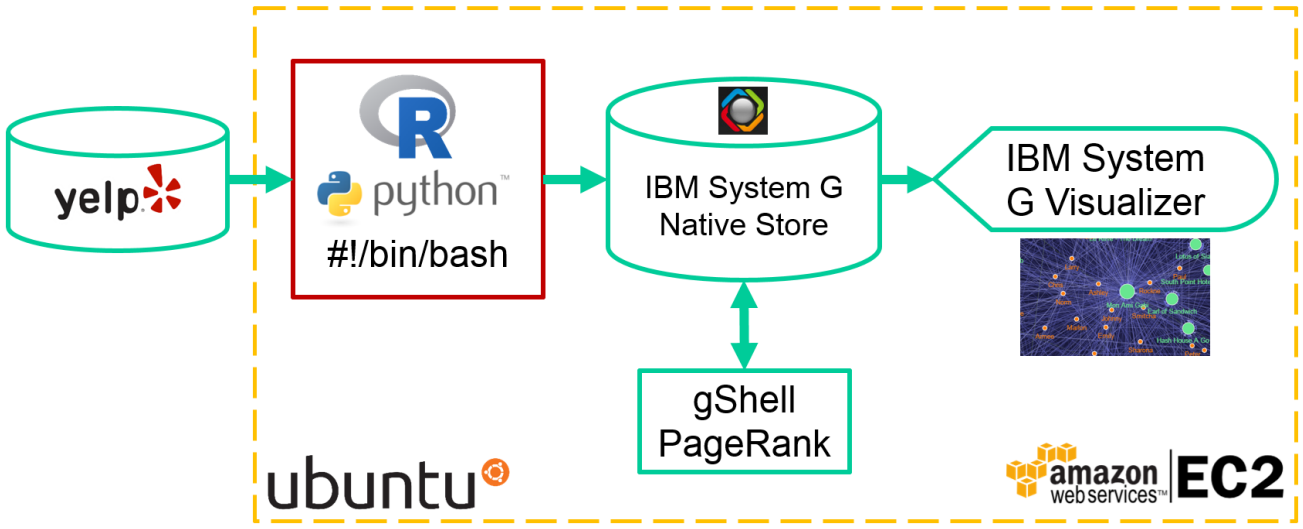

Figure 1. System Overview. The data came from Yelp as JSON files. These JSON files are processed using R, Python, and Bash into suitable vertex and edge files for IBM System G Native Graph Store. Rendering all 773k nodes and 2.7 million edges of the complete Yelp dataset would be unreasonable on a 1080p monitor, so subgraphs were created using gShell. The PageRank algorithm in gShell was used to calculate a value for node importance within the graph. Nodes and edges were rendered using the IBM System G Visualizer. All aspects of this project were hosted in Ubuntu 14.04 on an Amazon AWS EC2 cloud instance. The m4.large EC2 instance has 2 Intel Xeon 2.40 GHz E5-2676 processors, 8 GB of RAM, and 100GB of SSD storage.

recommender network. This visualization technique may help future researchers qualitatively evaluate their models in Yelp's network.

III. System Overview

An overview of the system can be seen in Figure 1. The Yelp Academic Dataset Round 8 contains data for businesses, reviews, users, check-ins, and tips in separate JSON files. The dataset also contains image files; however, image analysis is outside the scope of this paper. The 2.62GB dataset contains 2.7 million business reviews and 649k tips by 687k users for 86k businesses. The 687k users creates a social network with 4.2 million social edges. The structure of the business, user, and review JSON files can be seen in the Appendix : Yelp JSON File Structure.

IV. Algorithm

A. *Creating Edge and Vertex Files*

One of the major challenges of this project was converting the raw JSON files from Yelp into vertex and edge files for IBM System G. The JSON to CSV converter written in Python provided by Yelp had difficulty with escape characters in the JSON files. The converter worked for the user file, but did not work for the business and review files.

Attempting to replace the escape characters using python had mixed results. Some regular escape characters like "\n" for new line were easily replaced using the following python script.

```
with open("yelp_academic_dataset_review.json", "rt") as fin:
        with open("reviewFullOut.json", "wt") as fout:
                for line in fin:
                        fout.write(line.replace(r'\n', ''))
        fout.close()
fin.close()
```

However, other special Unicode "\u" escape characters still caused problems.

In order to fix this issue, I used the jsonlite library in R instead of Python to create CSV files for review edges and business vertices. The following R script creates the edge CSV file from the original review JSON file.

```
library(jsonlite)
reviews <- stream_in(file("./halfReviews.json"))
edges <- data.frame(
    source = reviews$user_id,
    target = reviews$business_id,
    type = reviews$type,
    date = reviews$date,
    stars = reviews$stars,
    votes = reviews$votes,
    review_id = reviews$review_id
)
write.csv(edges, "./eReviewshalf.csv", row.names=F)
```

The CSV edge file uses the user_id as the source of the edge and the business_id as the target. Date, stars, votes, and review_id are edge properties. The following R script creates vertex files from the original business JSON file.

```
library(jsonlite)
biz <- stream_in(file("./yelp_academic_dataset_business.json"))
biz$full_address <- gsub('\n',' ',biz$full_address)
nodes <- data.frame(
    id = biz$business_id,
    type = biz$type,
    name = biz$name,
    city = biz$city,
    state = biz$state,
    stars = biz$stars,
    latitude = biz$latitude,
    longitude = biz$longitude,
    review_count = biz$review_count,
    full_address = biz$full_address
)
write.csv(nodes, "./vBusinessFull.csv", row.names=F)
```

The business_id is the vertex id and the rest of the items are vertex properties. For the user vertices, using jsonlite and R to convert from JSON to CSV returned errors when trying to flatten the "friends" key. Therefore, I converted the raw JSON file to a CSV file using Yelp's Python script before selecting the relevant vertex columns with the following R script.

```
user <- read.csv("./yelp_academic_dataset_user.csv",head = T)
nodes <- data.frame(
    id = user$user id,
    type=user$type,
    name=user$name,
    review_count=user$review_count,
    average_stars=user$average_stars,
    yelping_since=user$yelping_since,
    fans=user$fans,
    votes.useful=user$votes.useful,
    votes.cool=user$votes.cool,
    votes.funny=user$votes.funny
)
write.csv(nodes, "./vUserFull.csv", row.names=F)
```

One disadvantage of this R script over the Python script is the RAM constraint. This script loads the entire JSON file into memory and converts it into an R data frame as a separate variable. This means that R is storing both the original JSON import and the converted data frame in memory during the conversion. Ubuntu's htop indicated that R used more memory during conversions than Python. To overcome this issue, I split the original JSON review file into multiple sections using Ubuntu's head and tail commands. For example, to make a review JSON file with 1 million reviews instead of 2.7 million, use the following shell command.

```
head -n1000000 input.json > output.json
```

B. *Relative Importance of Businesses with PageRank*

PageRank is a webpage ranking algorithm popularized by Google [12], [13]. The algorithm treats each webpage as a vertex in a directed graph and hyperlinks between webpages as directed edges. The algorithm outputs a probability that a random web surfer will end up at a webpage within the network of pages. The probability distribution over all vertices in the graph sums to 1. A damping factor incorporated in the algorithm accounts for random actions that the user might take. A damping factor of 0.85 means 85% of the time, the user will click on a link and go to a webpage linked to the user's current webpage and 15% of the time, the user will go to an arbitrary page.

Beyond webpage ranking, PageRank can be used as a network centrality measure where the PageRank value indicates the importance of a node [14]. For example, Kwak, Lee, Park, and Moon [15] applied PageRank to the Twitter social network to identify influential users.

In this paper, I used gShell's PageRank algorithm to identify influential businesses within a community. The PageRank of each business is the relative probability that a random business consumer will visit a certain business. This is the advantage of using PageRank over total review count to scale node size. Total review count does not take into account the constraint that a reviewer can only physically be in one business at a time. Since both users and businesses are nodes within the network, the raw PageRank gives a relative probability instead of an absolute probability. In order to get the absolute PageRank of only businesses, the PageRank of businesses must be normalized by the sum of only business node PageRanks.

V. Software Package Description

Instructions for setting up the environment as well as all code for this project is currently hosted at the following URL.

[https://github.com/Aneapiy/graph_visualization_Yelp]

A. *Data Processing Scripts*

The user and business JSON files were used to create vertex files. The python json_to_csv_converter.py provided with the dataset by Yelp converts the user JSON file into a CSV file. The R script makeVUserFull.R then uses that CSV user file as an input and outputs a user vertex CSV file called vUserFull.csv. The R script makeVBusinessFull.R creates a business vertex CSV file called vBuisnessFull.csv directly from the raw JSON file. If the user has 16GB of RAM or more, the R script makeEReviewFull.R can create the edge CSV directly from the raw JSON review file. If the user has less than 16GB of ram, the user needs to extract a subset of rows from the review file before running makeEReviewHalf.R. In all of the R scripts above, the input filename must match the user's files.

B. *IBM System G Native Graph Store*

IBM System G Native Graph Store stores the vertex and edge information directly as a graph instead of an indexed relational database with a graph front-end [11]. Using graph principles across all layers of the database allows System G to optimize graph operations with minimal overhead compared to other graph stores that use non-graph back-ends.

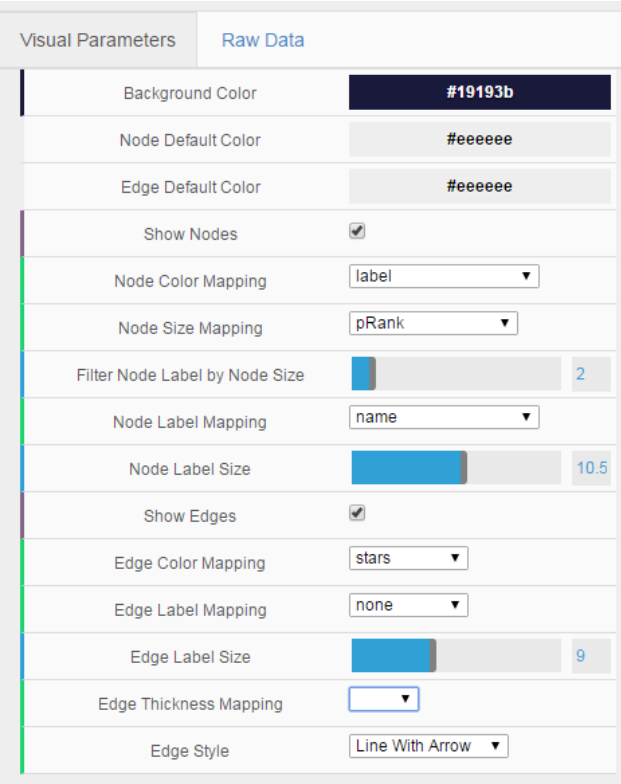

Figure 2: Visualizer settings. Setting the node color mapping as 'label' will make the graph display users and businesses as different colors. Using pRank (output of the PageRank algorithm) for node size mapping shows the importance of each business node. Since the network graphs in this paper do not contain social edges between users, all users will end up with the same sized node. Using stars as the edge color mapping provides a visual indication of how each user rated each business.

IBM System G uses the gShell interface to create graph stores. The gShell interface can be used in an REPL style shell by running the following command in the system G folder.

```
./bin/gShell interactive
```

The interface can also run a series of commands stored in a text file using the following command.

```
./bin/gShell interactive < cmds.g
```

The specific scripts for creating the graph store and subgraphs are in the github link above.

The commands for running gShell are contained in files with a ".g" suffix. In all of the provided gShell scripts, the user must change $FILEPATH to the user's specific path before running the command. The script for creating the full Yelp network graph is in createGraph.g. This script will load in all the vertices and edges processed previously by Bash, Python, and R. The repository includes two example scripts for creating a community subgraph for PageRank calculation. The script influentialBusiness.g creates a subgraph of all businesses that received >= 1,000 reviews and runs the PageRank algorithm. The script pittsburghGraphGen.g creates a subgraph of all businesses in Pittsburgh that received >= 100 reviews and runs the PageRank algorithm.

C. IBM System G Visualizer

The IBM System G Visualizer settings seen in Figure 2 renders a graph visualization similar to the user-business network graphs shown in Figure 3 and Figure 4 below. Since the visualizer runs on a web browser, the visualizer does have certain limitations on the number of nodes and edges that can be loaded at any given time. Through trial and error, I've found that the visualizer can load approximately 9,000 nodes and 20,000 edges comfortably. The graph store itself can support significantly more than that (millions of nodes and edges) since the graph store isn't constrained by the web browser interface.

VI. Experiment Results

To evaluate the methodology, I used the gShell script createGraph.g to generate a graph store called yelpFull. This network graph contains 771,527 nodes and 1,000,000 edges. This graph contains all users and businesses available in the Yelp Academic Dataset Round 8. The edges currently connect users to businesses with review data. No social edges between users are currently in the yelpFull graph. Rendering the full number of nodes and edges in the yelpFull graph store on a 1080p computer monitor would create too much clutter, so I used the gShell script influentialBusiness.g to create a sample subgraph for visualization. This subgraph called yelpPop, contains 8,826 nodes and 15,711 edges. A snapshot of yelpPop can be seen in Figure 3.

To visualize the influence of businesses within a geographical community, I used the pittsburghGraphGen.g script to create a subgraph called yelpPitt. The subgraph contains 8,331 nodes and 19,082 edges. Since this graph contains only businesses within Pittsburgh, the PageRank value of each business indicates how popular or influential each business is within the city. Businesses could use this type of geographically constrained network graph to study the influence and prominence of competing businesses. Users could also use a geographically constrained network graph to quickly identify popular businesses to visit.

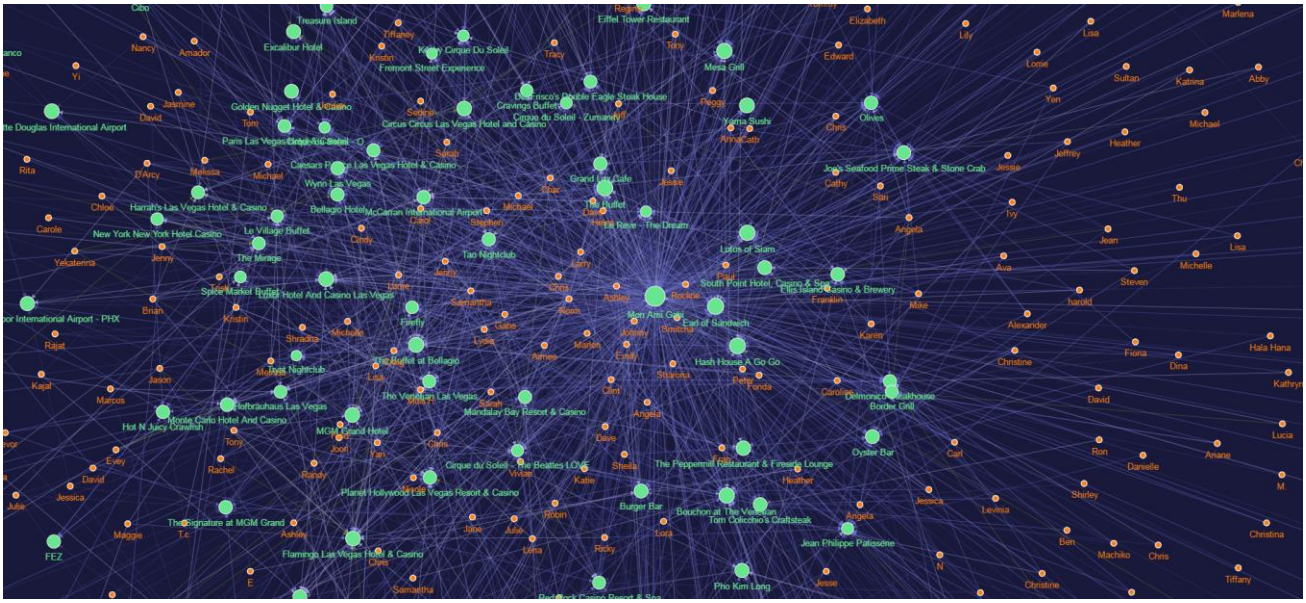

Figure 3: yelpPop network graph. The yelpPop network graph visualization above shows the egonet of "Mon Ami Gabi," the business with the highest number of reviews in the current Yelp dataset. Blue nodes are businesses while orange nodes are users. Edges between users and business indicate which users reviewed which businesses. The size of the nodes indicate the relative PageRank value of each node.

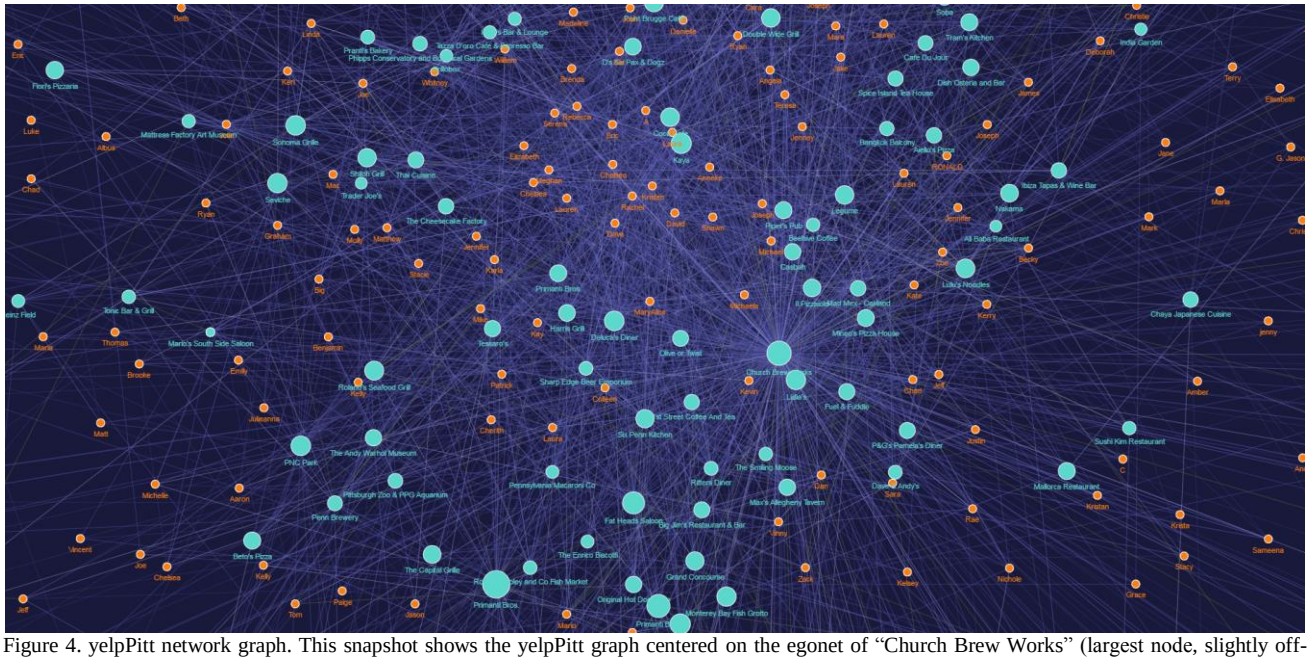

Figure 4. yelpPitt network graph. This snapshot shows the yelpPitt graph centered on the egonet of "Church Brew Works" (largest node, slightly off-center). Again, the size of the nodes indicate the relative PageRank value of each node. The difference in PageRank for these businesses are more pronounced than in Figure 3 above. This suggests that a smaller subset of businesses holds more influence within the Pittsburgh network.

VII. Conclusion

A. *Current Contribution*

The current Yelp network graph contains all users and businesses available in the Yelp Academic Dataset Round 8. Due to time and computation resource constraints, the graph contains 1 million edges between users and businesses instead of the 2.7 million available. The graph currently does not contain social edges between users. The two sample subgraphs yelpPop and yelpPitt demonstrates successful visualization of the Yelp social recommender network. By adjusting relative node size with the PageRank algorithm, the graphs display clear visual indication of which business

nodes held influence within the network. These graphs could provide researchers studying social recommender networks an alternative to word clouds for qualitative model evaluation. Yelp users could also use these graphs to quickly identify popular businesses within their area of interest. Subgraphs filtered by geography like yelpPitt could allow both businesses and users to identify businesses with high importance or influence within the immediate geographical area.

B. Future Work

To further refine the accuracy of the PageRank influence indication, I plan to incorporate the rest of the review edges along with the social edges between users. Incorporating social edges between users will allow the PageRank algorithm to not only show influential businesses, but also show key user nodes within the network. Identifying influential users will further improve the effectiveness of PageRank in identifying influential businesses.

Incorporating sentiment analysis of review text into review edges could be another area for further research. Running a Reverse PageRank analysis on these sentiment weighted review edges incorporated into the Yelp social network could show the origins of review sentiment within groups of users and businesses.

ACKNOWLEDGMENT

I would like to thank Professor Ching-Yung Lin and the IBM System G Graph Tools team for providing the graph store and visualizer. I would like to also thank TA Eric F. Johnson for providing a System G setup guide and the Ubuntu 14.04 AWS instance image.

APPENDIX : YELP JSON FILE STRUCTURE

Business:

```
{
  'type': 'business',
  'business_id': (encrypted business id),
  'name': (business name),
  'neighborhoods': [(hood names)],
  'full_address': (localized address),
  'city': (city),
  'state': (state),
  'latitude': latitude,
  'longitude': longitude,
  'stars': (star rating, rounded to half-stars),
  'review_count': review count,
  'categories': [(localized category names)]
  'open': True / False (corresponds to closed, not business hours),
  'hours': {
    (day_of_week): {
      'open': (HH:MM),
      'close': (HH:MM)
    },
    ...
  },
  'attributes': {
    (attribute_name): (attribute_value),
    ...
  },
```

User:

```
{
  'type': 'user',
  'user_id': (encrypted user id),
  'name': (first name),
  'review_count': (review count),
  'average_stars': (floating point average, like 4.31),
  'votes': {(vote type): (count)},
  'friends': [(friend user_ids)],
  'elite': [(years_elite)],
  'yelping_since': (date, formatted like '2012-03'),
  'compliments': {
    (compliment_type): (num_compliments_of_this_type),
    ...
  },
  'fans': (num_fans),
}
```

Review:

```
{
  'type': 'review',
  'business_id': (encrypted business id),
  'user_id': (encrypted user id),
  'stars': (star rating, rounded to half-stars),
  'text': (review text),
  'date': (date, formatted like '2012-03-14'),
  'votes': {(vote type): (count)},
}
```

REFERENCES

[1] Yelp. (December 19, 2016). *Investor Relations*. Available: http://www.yelp-ir.com/phoenix.zhtml?c=250809&p=irol-irhome

[2] Yelp. (November 1, 2016). *Yelp Dataset Challenge*. Available: https://www.yelp.com/dataset_challenge

[3] J. Huang, S. Rogers, and E. Joo, "Improving restaurants by extracting subtopics from yelp reviews," *iConference 2014 (Social Media Expo)*, 2014.

[4] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*, 2013, pp. 165-172: ACM.

[5] J. Linshi, "Personalizing Yelp Star Ratings: a Semantic Topic Modeling Approach," *Yale University*, 2014.

[6] N. Gupta and S. Singh, "Collective Factorization for Relational Data: An Evaluation on the Yelp Datasets," Citeseer2015.

[7] J. Wang, J. Zhao, S. Guo, and C. North, "Clustered layout word cloud for user generated review," *Yelp Challenge, Virginia Polytechnic Institute and State University*, 2013.

[8] F. M. F. Wong, Z. Liu, and M. Chiang, "On the efficiency of social recommender networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2317-2325: IEEE.

[9] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 137-146: ACM.

[10] F. M. F. Wong, Z. Liu, and M. Chiang, "On the efficiency of social recommender networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2317-2325: IEEE.

[11] I. S. G. Team. (December 20, 2016). *IBM System G Native Graph Store Overview*. Available: http://systemg.research.ibm.com/db-nativestore.html

[12] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," presented at the Seventh International World-Wide Web Conference (WWW 1998), Brisbane, Australia, 1998. Available: http://ilpubs.stanford.edu:8090/361/

[13] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web," Stanford InfoLab, Technical Report 1999, Available: http://ilpubs.stanford.edu:8090/422/.

[14] D. F. Gleich, "Pagerank beyond the web," *SIAM review*, vol. 57, no. 3, p. 321, 09/01 2015.

[15] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?," presented at the Proceedings of the 19th international conference on World wide web, Raleigh, North Carolina, USA, 2010.