Identifying Influencers in the Yelp Network

CS5344 Big-Data Analytics Technology - Group 18

Adithya Selvaganapathy
A0186084X

Chandrasekhar Sukumar
A0186109B

Dinesh Kumar Agarwal
A0186283W

MS Karthikeyan
A0186448N

Spatika Narayanan
A0088416X

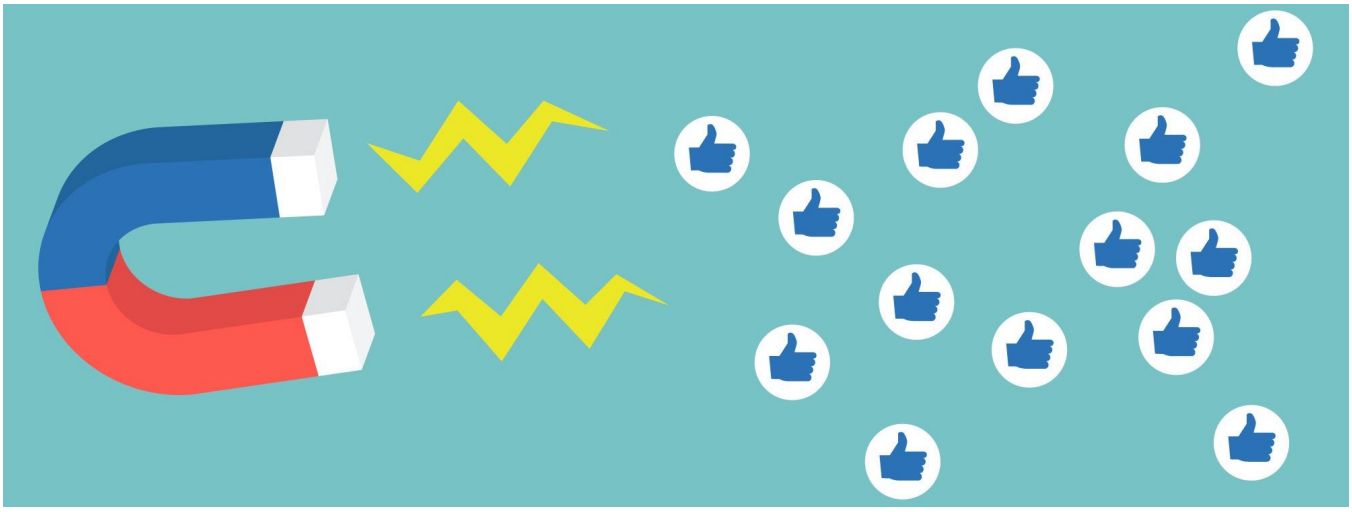

Figure 1: Social media reviews can influence consumer behaviour.

ABSTRACT

This report presents and compares methods for influence maximization in the context of the Yelp social network, using big data methodologies and algorithms. This work can be applied and extended to other social networks where it is relevant to identify influential users for targeted marketing and to maximize diffusion of information.

KEYWORDS

Big Data, Social Networks, Influence Maximization, PageRank

ACM Reference Format:
Adithya Selvaganapathy, Chandrasekhar Sukumar, Dinesh Kumar Agarwal, MS Karthikeyan, and Spatika Narayanan. 2019. Identifying Influencers in the Yelp Network: CS5344 Big-Data Analytics Technology - Group 18. In *CS5344 Big-Data Analytics Technology 2018/2019 Semester 2: National University of Singapore, Singapore*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CS5344 Semester 2, April, 2019, Singapore
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

1.1 Background

Yelp is a social media platform established in 2005, that connects people looking to try out a new restaurant or patronise a new business/service. It lists information about the business and a set of crowd-sourced user reviews posted by other members. Yelp has over 148 million reviews and about 77 million unique users access the platform every month.

1.2 Motivation

The influence of such crowd-sourced review content on consumer decision-making is growing significantly. With increased adoption of the Internet, the number of users and the content generated by them is growing manifold.

Since these reviews are typically not well-moderated and the volume of the data generated is huge, it is imperative to apply big data technologies to derive useful value from it. The Yelp data is apt for several graph mining opportunities; it provided us the opportunity to apply graph-based algorithms on a real-world dataset. Furthermore, success of a product depends upon the rate of adoption by innovators and imitators. A social media influencer is an innovator who has access to a large audience and can persuade others by virtue of their authenticity and reach.

A causal effect study using regression discontinuity analysis by Harvard Business School Assistant Professor Michael Luca [2],

revealed that a one star increase in a customer's Yelp review could translate to a 5 to 9 per cent increase in revenue for the concerned business.

It is easy to imagine, then, that businesses on Yelp could offer targeted discounts or deals to influencers, rather than offering a mass discount drive, for even greater returns.

Yelp has identified certain users as "Elite" reviewers based on the number of reviews they have written and/or their activity levels. This could be refined by giving Elite status only to those identified as influencers by virtue of their impact on other users' activities.

1.3 Problem Statement

Our aim is to use the user attributes and the reviews about the business units to identify the influencers on the Yelp network. We build the network by representing each user as a node and friendship between the nodes as bi-directional edges.

The influence of a given user, A, on B, is modelled as the directed edge weight from A to B. We use different ranking algorithms to order the nodes by their influence, and examine the distribution of those identified as top 100 by the algorithm, to Yelp's classification of the user as Elite or not.

2 DATASET

2.1 Schema

The dataset was made available by Yelp as part of Yelp Dataset Challenge [4] for educational or academic purposes. The dataset contains over 6.5 million reviews by over 1.6 million users covering 192k businesses spanning 10 metropolitan areas in North America. Of the available data, we used three of the JSON files in normalised form.

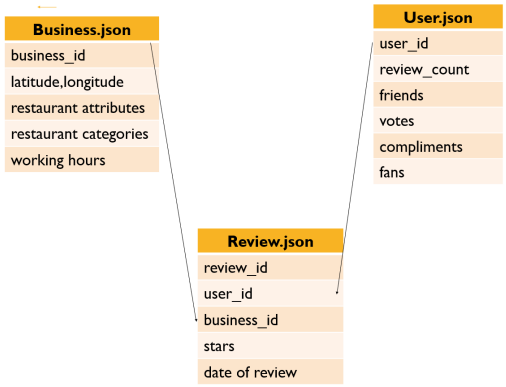

Figure 2: Schema of normalized data.

The Business table contained information about the restaurants, ratings, the attributes of the restaurants, the categories in which the restaurants are classified and business information like operating hours. The Review table contains information about the reviews written by an user for an business and information on how many people found the reviews funny, useful or cool.

The User table is the main table that contains information about the reviewer along with the friends list. This is the table based on which the network graph is generated by taking the user as a node, and his/her friends as other nodes connected through an edge.

2.2 Sub-setting

Since we are studying how a given user influences others to visit a restaurant or patronize a business on Yelp, it makes sense to study a single city in the dataset. This also ensures that we have a denser (rather than sparser) graph of a single community. To choose the city of study, we performed some visualizations and exploratory data analysis.

The distribution of user reviews, business reviews and user profiles were analysed based on the city they are associated with. As we can see from the yellow pie segments in Figure 3, users of Toronto were very active as the proportion of user reviews (7.4) was almost double the user percentage (4.6).

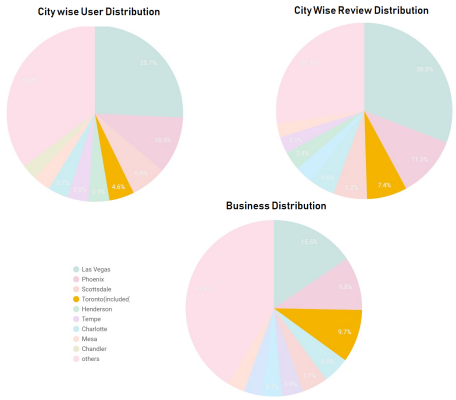

Figure 3: Distribution by city.

A quick analysis was also done on the quality of the reviews posted by the users in Toronto. It was observed that the contribution of positive reviews were high (Figure 4). Since we are only interested in reviews with a positive influence on users, only those with more than a 3-star rating were retained.

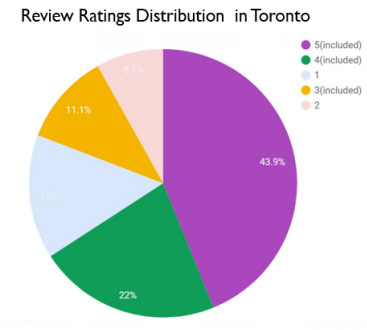

Figure 4: Distribution of ratings.

After removing these negative reviews, the distribution of number of friends were analysed in four different buckets (Figure 5)

ranging from 0 to 100, less than 250, less than 500 and greater than 500. Around three-fourth of the users have only less than 100 friends.

Even considering just positive reviews in a single city, we are left with a large graph of around 11,600 nodes and 172,588 edges. As a first step, we filtered the graph to only the users having more than 500 friends. There were around 708 such users, and the graph reduced to 21,765 edges. We later compare the results of our ranking algorithms on these two graphs.

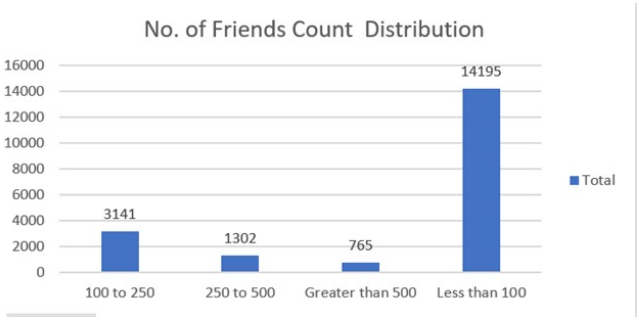

Figure 5: User classification based on no. of friends

3 LITERATURE REVIEW

Influence maximization is the optimization problem of finding a small set of seed nodes, say, S, which could maximize the spread of information in a network. Social influence analysis is commonly studied using the independent cascade (IC) and linear threshold models (LT). [6]. For both these models, this problem has been proved to be NP-hard - i.e. no known algorithm can solve it in polynomial time [5–7]. However, as these sources also note, there are heuristics or greedy algorithms that can be applied to these models to produce a solution.

For example, Luo, Z. et al. [7] proposed a PageRank-based heuristic - i.e. restricting the seed nodes to just those with high PageRank. For our present work, this is an indicator that PageRank can be used as an initial influence measure, since implementing the entire heuristic greedy algorithm proposed is out of scope.

Further, as Li, K. et al. [6] note, IC and LT models require a graph with edge weights reflecting the influence as input. So there needs to be a way to quantify the influence. In [1], the authors have quantified influence in Twitter, by considering the temporal aspect of tweets. That is, if B is following A, and B tweets a URL after A has, then they consider A to have influenced B. In case, B is following multiple people who have tweeted the same URL before, they [1] assign the influence to just the first person, assuming "first influence": "individuals are influenced when they first see a new piece of information, even if they fail to immediately act on it."

In the present work, we quantify influence similarly using the temporal aspect of reviews, as we describe in the next section. However, the credit is assigned independently: it is neither split between nodes, nor does it go to solely the first or last influencer.

4 METHODOLOGY

4.1 Preprocessing

The following preprocessing steps were executed on the input JSON files using Google Big Query:

(1) Combine different types of votes (e.g. useful, funny) into total_votes.
(2) Combine different types of compliments (e.g. cute, cool) into total_compliments.
(3) Removed reviews having 3 stars or less.
(4) Subset for business units present only in the city of Toronto.
(5) Removed users who have written less than the average number of reviews.
(6) Removed users having lesser than 500 friends, for faster processing and to compare with results on the whole Toronto graph.

4.2 Graph Creation

Our aim is to convert the un-directed input from Yelp into a directed graph with the edge weights representing the influence factor.

Input from Yelp:

The adjacency (friends) lists: A is in B's friends list, and vice-versa.

Transform to edge list and calculate edge weights:

A is a friend of B and has an influence of 0.5 on B.
B is a friend of A and has an influence of 0.7 on A.

Figure 6 below elucidates our approach:

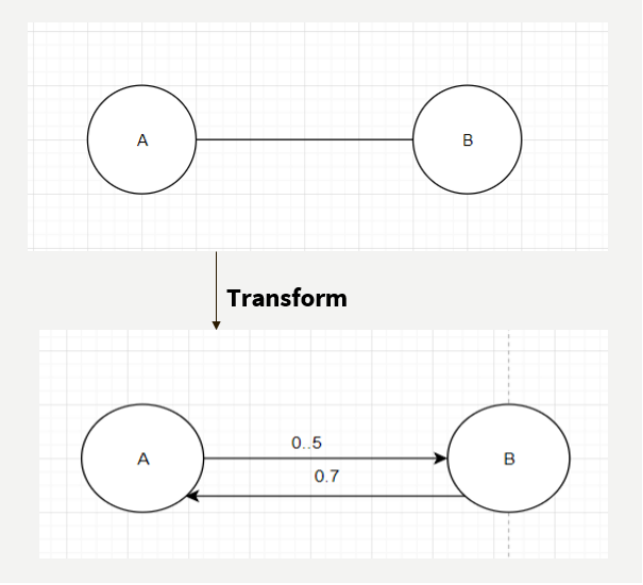

Figure 6: Input graph transformation.

Our output graphs (as mentioned in Section 2.2) are the following:

- Graph 1: Filtered edges - 21765, Nodes - 708
- Graph 2: Edges - 172588, Nodes - 11609

4.3 Edge Weight Calculation

Edge weights which represent the influence of A on B, for directed edge A to B, are measured on a scale from 0 to 1.

The edge weight calculation involves two components:

(1) Components dependent on source node profile:
- **Fans Ratio:** Fans of A/Max. number of fans
- **Votes Ratio:** Votes received by A/Max. number of votes
- **Compliments Ratio:** Compliments received by A/Max. number of compliments

The maximum number of fans/votes/compliments is the the maximum number of fans/votes/compliments present in the dataset as a whole.

(2) Components dependent on destination node profile:
- **Following Ratio:** Number of reviews written by B after a review of the same restaurant by A/Total number of reviews by A

Our assumption is that the Following Ratio could act as a proxy for the number of restaurants B visited on seeing A's positive review.

Our final edge weights formulation can be represented as:

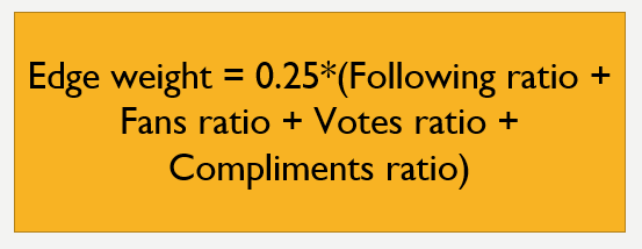

Figure 7: Formula for edge weight calculation

4.4 Model Output and Evaluation

(1) Use the ranking algorithms to identify the users with highest influencing potential.
(2) Take the top 100 influencers identified and calculate the percentage of those in the Yelp Elite squad.
(3) The best algorithm would be the one which identifies maximum number of users who are *not* in the Elite squad.

We look at the number of users who are *not* Elite as our metric, since Yelp's current strategy is based on the user activity like the number of reviews, friends and fans. It may be helpful for them to find influential users through other attributes like review behaviour (as we present in this work), since they are ostensibly missing out on these users who could potentially be *made* appropriate Elite members.

Though the number of friends is an important indicator of popularity, and hence, influence, it doesn't account for the quality of the connections. For example, a user may have only 10 friends, but all of them may be influential or expert critics, making the original user influential in turn - a positive review by him/her could attract the attention of others which could make/break the reputation of a restaurant.

5 TOOLS AND TECHNOLOGIES

The challenge in solving this problem was the variety of tools we needed to set up and use, as there is no single package that can be leveraged to apply the different ranking algorithms we wanted to study.

- BigQuery for data preprocessing and generating final edge list.
- Apache Spark, and its graph processing library, GraphFrames, for PageRank and Naïve Influence Identification.
- Python's NetworkX package for Weighted PageRank.
- The Independent Cascade add-on to NetworkX adapted from GitHub. [3]
- IntelliJ IDE and Jupyter Notebook as the development environments.
- Gephi as the visualisation software.

6 ALGORITHMS

6.1 Naïve Influence Maximization

For the baseline level of influencer identification, we work on the simple assumption that if two Yelp users are friends, then they both exert influence on each other.

By that assumption, the top influencers should be those with the biggest neighbourhood. We have measured the size of a node's neighbourhood in two ways.

6.1.1 One-hop Neighbours: First, we count the number of one-hop neighbours - or number of friends for each node. With the constructed graph on GraphFrames, this is simply the number of out-degrees for each node.

Alternatively, since our graph is symmetric (i.e. every user who is a destination node, is also a source node) and bi-directional, we could also count the number of in-degrees. This is fairly simple using GraphFrames' API.

6.1.2 Two-hop Neighbours: The second method of neighbourhood measurement, is to count the number of two-hop neighbours. This includes Yelp users that are direct friends of a node, *and* all the friends of those direct friends.

Doing this iteratively would be computationally intensive. Instead we leverage GraphFrames' motif-finding. Network motifs are structural patterns that occur repeatedly and represent the relationships between nodes and/or edges. Here, we use it to identify all edges that fit the following structural pattern:

$$[(A) - [x1] - > (B)] \wedge [(B) - [x2] - > (C)] \wedge \neg[(A) - [] - > (C)]$$

That is, when counting C as a two-hop neighbour of a given node A, there must be an edge from A to B, and B to C, but no direct edge from A to C, since we separately add the number of direct friends of A.

However, simply measuring the size of the neighbourhood, doesn't account for the *importance* of the nodes in A's neighbourhood.

6.2 PageRank

We don't just want to look at absolute size of the neighbourhood when considering a given node. We also want to look at the "quality" of the links to A, which is where PageRank comes in.

The more influential users that a user A, is friends with, the more influential they should be considered. As noted in Section 3, and in [7], PageRank can be an important indicator of influence maximization.

Again, input to the algorithm is the directed graph with Yelp users as the nodes, and friendship as a bi-directional edge.

The drawback of this approach, is that we still consider only 'friendship' as influence, and haven't looked at other aspects of a Yelp user's profile or review activity.

6.3 Weighted PageRank

The Weighted PageRank algorithm was first proposed as an extension to the standard PageRank by Wenpu Xing and Ali Ghorbani[8]. So far, we haven't considered user attributes or review behaviour. This can be done by considering the influence exerted by a node on its surrounding nodes, as we calculated for the edge weights previously.

So, we use this modification to PageRank so that votes from a given node are distributed in proportion to the weights of its edges, instead of equally.

In our problem, we have shown the edge-weight calculations in Section 4.3.

6.4 Independent Cascades

In the independent cascading model we have a directed graph G. If we have an un-directed graph, we change it to a directed graph. We have an initial set of nodes which start the diffusion process.

When any node is activated, it will have only one chance to activate any of the neighbouring nodes, that are currently inactive. This activation attempt is successful in proportion to the edge weights (called as propagation probabilities). This edge weight basically tells the probability with which one node influences its neighbouring node. This diffusion process happens in a distinct manner.

Once this activation attempt is successful, the node will become active in the next step. Any activation attempt from one node to next can only happen once even if the attempt fails. Also, the activation attempt on the node from each of its incoming links is independent of any other previous attempts made to activate that node. This procedure will continue until the stage when any more activation attempt is not possible.

The objective function for this influence maximisation problem is to find the initial activation S for which the influence spread (the expected number of active nodes at the end of the diffusion process) is maximum.

This problem has been defined by Kempe et al., [5]. As noted in Section 3, this influence maximization problem is NP (non-deterministic polynomial) hard for independent cascading modes.

But as the objective function (expected influence spread) is a monotone and sub-modular function, we can use a greedy algorithm. That is, we can select any new node that will give the highest marginal gain in the objective function and approximate the result based on Monte-Carlo simulations.

In our problem, we have taken the propagation probabilities based on the edge weights calculations defined in previous section.

7 VISUALIZATION

After completing the modelling, it is important to identify the visualization method in which the output will be delivered to the end-user. The advantage with graph data is that the the relationship represented by the nodes and edges is quite interpretable. The software used for creating the visualisation was Gephi. Gephi is a visualisation and exploration software for graphs and networks.

The software has built-in algorithms like PageRank, HITS and community detection. The node size can be adjusted based on the PageRank scores calculated. Also, different colour encodings can be used for nodes belonging to different communities. A sample graph containing Yelp data with 50 nodes and 415 edges is shown below. The node size is adjusted based on its PageRank value, to easily identify the user with highest rank.

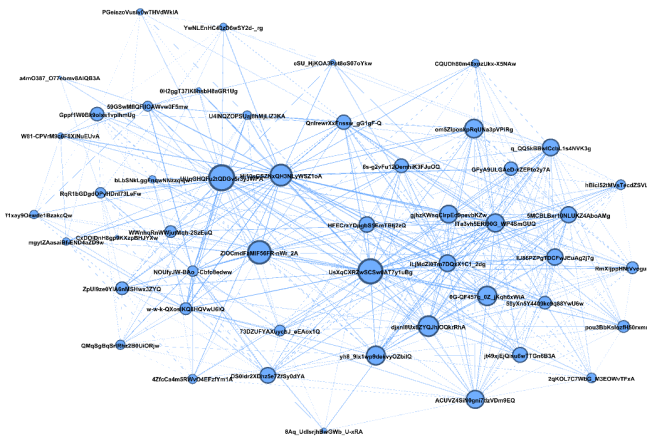

Figure 8: 50 of the Yelp network users with PageRank.

This tool can help business managers take decisions on campaigns or promotions and identify the best strategies yielding optimal results.

8 RESULTS AND ANALYSIS

As mentioned in Section 4.4, to measure the performance of our algorithms, we looked at the percentage of Elite users in top 100 users ranked by the various algorithms. In the dataset provided by Yelp, users are classified as *Elite* based on the years when they were given Elite member status. A user can be given Elite status for multiple years.

For simplicity, we consider a user as Elite if they were awarded Elite status in at least one year. We compared the top 100 users ranked by our algorithms with Yelp's Elite classification.

The results based on the different algorithms are given in Tables 1 and 2, for Graph 1 and Graph 2 respectively.

We can see from the results for the two naïve algorithms in Tables 1 and 2, that Yelp is probably giving more weight to users having more friends. With these algorithms, a higher percentage of the top 100 are Elite users. But we see that with algorithms such as Weighted PageRank and Independent Cascades, where we factor

Table 1: Comparison of Algorithms - Filtered Edges

Algorithm	Top 100 Elite Users	"New" Elites
One-hop Neighbours	99	1
Two-hop Neighbours	98	2
PageRank	99	1
Weighted PageRank	97	3
Independent Cascades	98	2

Table 2: Comparison of Algorithms - No Filter

Algorithm	Top 100 Elite Users	"New" Elites
One-hop Neighbours	97	3
Two-hop Neighbours	99	1
PageRank	94	6
Weighted PageRank	95	5
Independent Cascades	96	4

in other features from the dataset (through the edge weights) the percentage reduces.

With two-hop neighbours (Table 2) there were 99 matches, whereas with Independent Cascades we found 4 users who were not classified as Elite by Yelp but were in the top 100. On examining their attributes in the original dataset, we see that these 4 users had a higher proportion of reviews which were rated more useful. So we can theorize that Yelp may be losing out on these 4 users who are influential, by not having categorized them as Elite.

When comparing the results from Table 1 with Table 2, which contains results from the full Toronto graph, we can also see that, on average, Table 1 algorithms identify more Elite users. This makes sense since in Table 1, the graph under study is Graph 1, with all users having more than 500 friends. So by Yelp's metrics/naïve influence identification, most users in the graph would already be Elite users, with very few non-Elite users left to identify. So this result is expected.

9 FUTURE WORK

There are some enhancements which can be done to improve our analysis and broaden the scope of our study.

9.1 Time-based Edge Weights

While calculating edge weights we are assuming that, if a user reviews a restaurant and his friend reviews an restaurant, the user's review influenced his/her friend into making this decision. We are not considering the time effect in the edge weight calculation.

For instance, the friend could have written the review, say, 6 months after the original user has written the review. The timeframe can be taken into consideration by which different (decaying) weights can be assigned from 1 to 0 based on the time lapse. This would give higher weights to a smaller time-lapse and help in measuring the influence factor of different friends more effectively.

9.2 Negative Effect

In the model we have taken a subset of the reviews, where all ratings are positive. There could be negative reviews by the influencers which would have resulted in his friends not visiting the restaurant. This would be more difficult to measure as it would be difficult to find a reason why a visit was not made. To find whether a review is positive or negative we can calculate sentiment scores using Python packages such as 'afinn', and use it for edge weight calculation.

9.3 Clustering

We can use clustering algorithms to find clusters within the network. The identified clusters can be used to run campaign programs within the user community. For example the community may have some common attributes based on cuisines that its members enjoy, like Indo-Chinese, or may be based on facilities available in the restaurant like sports bar etc. This community detection can help in maximising the results of campaigns and promotions by the restaurants, and would be a good complement to just simple influencer identification.

10 CONCLUSION

We have seen how different attributes can be used to quantify influence - e.g. number of friends/reviews, and review behaviour/timing. We have studied different graph analysis algorithms in the context of influence measurement or maximization. We have explored the use of different graph visualization and/or analysis tools and packages. We have learned nuances of handling big-data using a distributed processing framework.

We saw how the use of these big-data technologies can help to unlock business value by deriving meaningful insights. Combining these tools, with machine learning models and the features elaborated on in Section 9, will help in finding actionable insights and would enable making data driven decisions.

REFERENCES

[1] Eytan Bakshy, Jake M Hofman, Winter A. Mason, and Duncan J. Watts. 2011. Identifying Influencers on Twitter. In *Fourth ACM International Conference on Web Seach and Data Mining (WSDM)*.

[2] Michael Blanding. 2011. The Yelp factor: Are Consumer Reviews Good for Business? *Boston: Harvard School of Business* (2011).

[3] Hung-Hsuan Chen. 2013. NetworkX Add-on. https://github.com/hhchen1105/networkx_addon/blob/master/information_propagation/independent_cascade.py.

[4] Yelp Inc. 2019. Yelp Open Dataset. https://www.yelp.com/dataset

[5] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the Spread of Influence through a Social Network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 137–146.

[6] Kan Li, Lin Zhang, and Heyan Huang. 2018. Social Influence Analysis: Models, Methods, and Evaluation. *Engineering* 4, 1 (2018), 40–46.

[7] Zhi-Lin Luo, Wan-Dong Cai, Yong-Jun Li, and Dong. 2012. A PageRank-based Heuristic Algorithm for Influence Maximization in the Social Network. In *Recent progress in data engineering and internet technology*. Springer, 485–490.

[8] Wenpu Xing and Ali Ghorbani. 2004. Weighted PageRank Algorithm. In *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.* IEEE, 305–314.