Analyzing LinkedIn Job History and the Job Migration Graph

Problem Definition

As a software engineer working in the San Francisco Bay Area, there are many great options of places to work after college. It's not uncommon for a good engineer to have several great offers to choose from when deciding on a new job. But all else being equal -- pay, responsibility, team fit -- how does one decide which company will provide the most long-term value for one's career?

The proposal of this project is to take well-known graph ranking algorithms applied to other domains -- most notably the PageRank algorithm used in web search -- and apply it to a company graph data set -- where an edge from company A to B denotes that an employee of company A left the firm to join company B. We then explore the idea of tuning the PageRank algorithm to fit extra information about job hops that we can infer from the data, and examine how these tweaks affect the ranking of the different companies.

For example, how does an employee's tenure with their previous employer affect the ranking of their current employer? All things being equal, we intuitively believe that it says more about a company's prestige if their new hire was with their previous employer for fifteen years as opposed to one. How does the size of a company affect its ranking, and can we somehow adjust for this in our formulation of the PageRank? Because job transitions exist at a particular point in time, should we not give higher weight to more recent edges for determining a company's rank in the present?

Because there is little concrete data about company rank and prestige, most of these questions cannot be answered conclusively, and instead come down to value judgements in some part. In this study, we'll explore each of them in turn and measure the effect on the company rank, and attempt to analyze these effects algorithmically in a manner similar to the work done by Page and Brin in their original paper.

Related Work

There are a handful of proprietary company rankings published on the Internet. LinkedIn publishes a list of so-called "Top Attractors"[1], companies that job seekers most actively pursue for employment. Silicon Valley salary analytics company Paysa released a similar score called

[1] https://lists.linkedin.com/2016/top-attractors/en/us

"Company Rank"[2] that claims to quantity the density of tech talent within a company. The results from these studies are interesting, but there's little said with respect to their methodologies.

Quartz[3] published an article in 2015 showing where top tech companies recruit their talent from, using data derived from LinkedIn, but without the raw dataset to verify and expand on the results.

Academic work on this subject is limited, but not entirely absent. One student in Waterloo, Canada did a study on 10,000 local tech companies using machine learning ranking techniques[4]. Isaac Sorkin from the University of Michigan did an extensive study in 2015 on the topic of firm ranking[5]. The author mentions briefly the idea of applying PageRank to this problem, but focuses primarily on the relationship between compensation and employee migration between firms.

Data Collection

Data has been collected by scraping public profiles off of LinkedIn. To date we've collected 86,908 positions across 17,084 user profiles. The works out to a graph of 34,910 company nodes with 40,999 job transfer edges. Notably, only the first 6154 companies have more than one employee in the dataset. Currently there's no additional sanitization to remove companies with low representation from the data, but that's something we hope to explore later.

Top 10 Companies by Profiles in the Dataset

Rank	Company	Profiles
1	Google	1462
2	Microsoft	1020
3	Amazon	699
4	Facebook	672
5	Qualcomm	644
6	LinkedIn	546
7	Apple	484

[2] https://www.paysa.com/company-rank
[3] http://qz.com/342229/where-tech-companies-hire-from/
[4] https://github.com/shinew/tech-company-ranking
[5] https://sites.google.com/site/isaacsorkin/papers/sorkin_revealedpreference.pdf

8	Yahoo	412
9	Intel	377
10	Uber	358

The data collection process was intentionally biased towards software engineers in the San Francisco Bay Area to limit the scope of the study. When crawling the data, we began at a random page of a software engineer in the SF bay area, and performed a breadth-first search by exploring outward via the “People Also Searched For” panel on the page. We only added a link to our queue if the profile header contained one or more keywords indicating that the profile likely belonged to a tech employee (e.g., “software”, “engineer”, “data”, “developer”). In total, 6282 of the positions in the dataset were for “Software Engineer”, and 4883 of the positions were based out of the San Francisco Bay Area, the most represented values for their respective categories. It should be noted that most positions had no location provided, explaining the low representation of the bay area across all profiles.

Ideally, we would have liked to collect significantly more data, and have the luxury of further sanitizing it to ensure that only tech employees are represented in the sample. However, we were very careful in the crawling process to ensure that we didn’t exceed our request limit from LinkedIn, which significantly slowed the process of data collection down.

Methodology

We began by employing the basic PageRank algorithm on the dataset, with a teleport probability $\beta = 0.85$ and multiple edges being counted repeatedly:

$$PR(u) = \frac{(1-\beta)}{N} + \beta \sum_{v \varepsilon B_u} \frac{L(v, u)\, PR(v)}{L(v)} = \frac{(1-\beta)}{N} + \beta \sum_{v \varepsilon B_u} \sum_{(v->u) \varepsilon E} \frac{PR(v)}{L(v)}$$

Where $PR(u)$ is the PageRank of node u, B_u is the set of all nodes linking to u, and $L(v)$ is the number of edges going out of node v, and $L(v, u)$ is the number of edges going from v to u. In the alternative formulation, E is the set of all edges in the graph. We count multiple transfers from company A to B repeatedly to account for the intuition that a lot of people going from A to B is a stronger signal or endorsement than a few people going from A to C.

The following table shows the PageRank for all companies using no additional weighting, along with their change in rank from the previous table of most represented companies by profiles (delta). Our assumption was that without any re-weighting or normalization, the PageRank would correlate heavily with the number of profiles associated with the companies.

Top 10 Companies by PageRank in the Dataset

Rank	Company	PageRank	Delta
1	Google	0.02220	0
2	Facebook	0.01263	2
3	Uber	0.00963	7
4	Microsoft	0.00962	-2
5	LinkedIn	0.00960	1
6	Apple	0.00719	1
7	Amazon	0.00681	-4
8	Qualcomm	0.00535	-3
9	Twitter	0.00441	6
10	Carnegie Mellon U	0.00406	3

In general, companies with a lot of employees have higher PageRank, though there are some exceptions. Texas Instruments dropped 40 places to rank 63 in the PageRank, while Dropbox managed to jump 50 places to rank 30. Pinterest jumped 55 ranks, and Slack jumped 129 ranks.

We performed extensive experimentation with different modifications to the PageRank formulation to improve the results. One modification was to add an edge from A to B for every year spent at company A before transferring to B. This had little effect on the PageRank results, but did manage to push AirBnB from rank 13 to rank 10.

A more impactful change was to add a recency bias, so that one's current position has significantly more weight than one's previous positions. As a rough realization of this, we added 100 additional edges from A to B is B is the user's current company. This managed to push high profile unicorns Uber to rank 2 and AirBnB to rank 7, and further pushed down older more established companies such as Qualcomm, IBM, and Oracle.

These experiments provided interesting results, but didn't completely address the most fundamental issue with vanilla PageRank: the bias towards company size. In general, large companies with a lot of employees tended to rise towards the top of the ranking, and smaller companies tended to be pushed down for the same reason. Intuitively, we expect that there

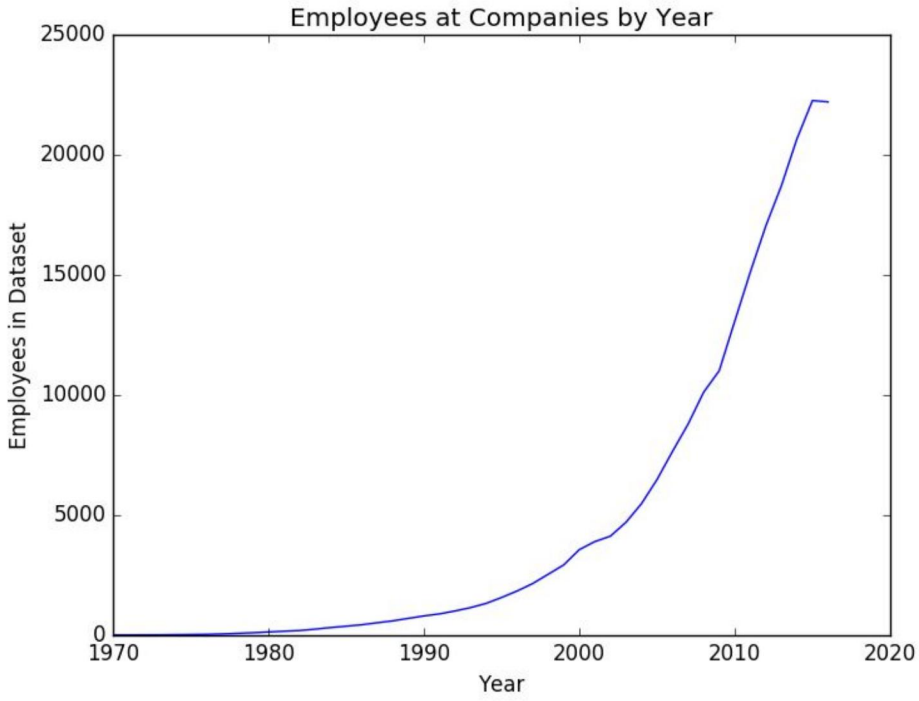
would be many prestigious startups with a highly selective screening process that would end up lower in the rankings just by virtue of having a low number of absolute in-links.

To address the bias of company size, we created our final variation on PageRank we've termed **Size Down-Weighted PageRank**, that applies more weight to links going to companies with a fewer number of total employees in the year of the job transfer.

Concretely, let $\delta_{c,y}$ be the observed number of employees at company **c** in year **y**, and let $\delta_y = max_c\, \delta_{c,y}$ be the maximum number of observed employees at any company in year y. Let O_v be the set of outgoing edges from v. Then we compute the weighted PageRank score for a company to be:

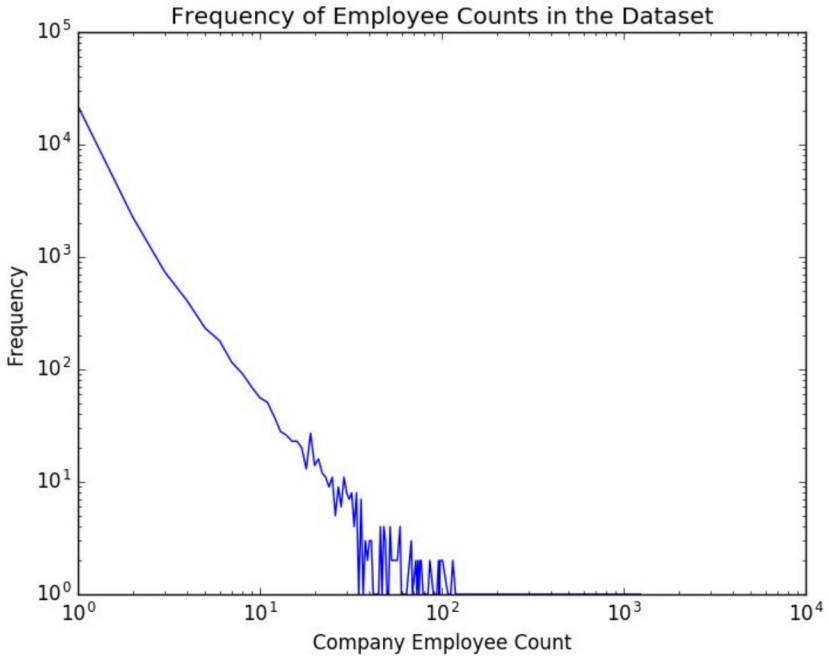
$$PR(u) = \frac{(1-\beta)}{N} + \beta \sum_{v \varepsilon B_u} \sum_{(v->u) \varepsilon E} \frac{\delta_{y_{(v->u)}} - \delta_{cu,y_{(v->u)}} + 1}{\sum_{w \varepsilon O_v} \delta_{y_{(v->w)}} - \delta_{cwy_{(v->w)}} + 1} PR(v)$$

In practice, we compute this result by iterating over every job transfer in the dataset, and adding an additional link to the graph from company A to B for the difference $\delta_{c,y} - \delta_y$. From there we can run the vanilla PageRank algorithm over this reweighted graph to obtain our altered ranking vectors.



A few additional data sanitization steps were done to improve on the previous ranking results. First, we note that Carnegie Mellon University had the 10 rank position using the vanilla PageRank algorithm. Upon digging into the data, it became clear that this was due almost exclusively to interns at Big 4 tech companies going back to be research assistants as CMU. We removed any work history at and prior to internships (by filtering positions containing the case-insensitive string “intern”). The net effect was to completely remove universities from the top levels of the rankings, and to give a slight boost to the unicorn startups (Uber, AirBnB, Pinterest, Slack, Square) that take significantly fewer interns each year.

Another issue with the data is the skew towards “companies” with only one or a handful of employees represented. When looking at the distribution of companies by number of employees that worked there at any point in time, there was significant skew towards the single-digits. We removed all links in the graph to companies with fewer than 100 employees.



Using this approach, we were able to compute the following top 10:

Top 10 Companies by Company Size Down-Weighted PageRank

Rank	Company	PageRank	Delta
1	Google	0.06344	0
2	Uber	0.05625	6

3	Facebook	0.04747	2
4	LinkedIn	0.02376	2
5	AirBnB	0.01716	16
6	Twitter	0.01510	8
7	Microsoft	0.01432	-5
8	Apple	0.01369	-1
9	Amazon	0.00985	-5
10	Netflix	0.00914	20

Looking at the data we see that the largest benefactor of the company size bias -- Qualcomm -- has been pushed completely out of the top 10 ranking, and smaller but more popular unicorn startups like AirBnB have been significantly pushed up.

Conclusion

In comparing our results to other rankings of Silicon Valley tech companies, we found significant overlap:

Top 10 Companies by Varying Ranking Algorithms

Rank	**Size Down-Weighted PR**	**Paysa CompanyRank**	**LinkedIn Attractors**
1	Google	Uber	Google
2	Uber	Google	Salesforce
3	Facebook	AirBnB	Facebook
4	LinkedIn	Pinterest	Apple
5	AirBnb	Facebook	Amazon
6	Twitter	Dropbox	Uber
7	Microsoft	Amazon	Microsoft
8	Apple	LinkedIn	Tesla
9	Amazon	Salesforce	Twitter

10	Netflix	Square	AirBnB

Comparing the Paysa top 10 to our positions:

Top 10 Companies by Paysa vs Our Methodology

Rank	Paysa CompanyRank	Size Down-Weighted PR Ranking
1	Uber	2
2	Google	1
3	AirBnB	5
4	Pinterest	N/A
5	Facebook	3
6	Dropbox	N/A
7	Amazon	9
8	LinkedIn	4
9	Salesforce	N/A
10	Square	N/A

6 of the top 10 companies in the Paysa CompanyRank were in the top 10 of our weighted PageRank. The 4 companies missing from the top 10 of Paysa's ranking all had fewer than 100 total profile samples in our dataset. It's reasonable to assume that with more samples, these companies would be closer to the Paysa ranking equivalent.

Of the 26677 companies in the final ranking, all 10 were within the top 100 when filtering out to only companies with fewer than 10 employees. Pinterest was the highest, at rank 19. The lowest of the Paysa top 10 in this dataset -- Dropbox at rank 63 -- had only 43 profiles that worked at that company at any time. Empirically, companies with fewer than 50 to 100 profile samples tended to have rankings that were unstable, and subject to over weighting due to a handful of employees that appeared in the dataset. While Pinterest, Dropbox, Salesforce, and Square all appeared lower than expected with only a handful of samples, The Apache Software Foundation show up to rank 10 with only 43 samples.

We conclude that it's simply not statistically meaningful to include companies with fewer than 100 samples in the results.

Similarly for the LinkedIn top attractors:

Top 10 Companies by LinkedIn vs Our Methodology

Rank	LinkedIn TA	Size Down-Weighted PR Ranking
1	Google	1
2	Salesforce	N/A
3	Facebook	3
4	Apple	8
5	Amazon	9
6	Uber	2
7	Microsoft	7
8	Tesla	N/A
9	Twitter	6
10	AirBnB	5

This time, 8 of the top 10 attractors appeared in our top 10, and all the top 10 LinkedIn Top Attractors were within our top 100 when filtering our companies with fewer than 10 employees. Comparing Paysa to LinkedIn directly, we find that they share six of their top 10 companies: Google, Uber, AirBnB, Facebook, Amazon, and Salesforce. 4 companies only appear in Paysa's ranking: Pinterest, Dropbox, LinkedIn, and Square. 4 companies only appear in LinkedIn's ranking: Apple, Microsoft, Tesla, Twitter.

The only company represented in both Paysa and LinkedIn's top 10 but not ours was Salesforce, attributed to low representation in the dataset, with only 53 samples. Conversely, the only company represented in our top 10 but neither Paysa nor LinkedIn's is Netflix. Netflix was right on the edge of the cutoff at 101 samples. It ranked 21 on Paysa's CompanyRank and 11 on LinkedIn's Top Attractors, so this discrepancy is not surprising.

Overall, we believe that Size Down-Weighted PageRank performs well at accurately mapping job transfers to company endorsements using the same intuition as PageRank applied to the web, but additionally corrects for large variations in company size that would otherwise result in big companies absorbing all the PageRank. The accuracy of the results becomes unstable as the number of profile samples drops below 100, but with additional data these results can be

made even more accurate and give even more weight to small but prestigious unicorn startups in the Bay Area.