Job Hop Network Analysis

A. Mazzetto

December 2023

1 Introduction

This study is concerned with the analysis of a talent flow network. These are directed graphs where edges represent job hops and nodes might be job titles (*job role network*) or firms (*job network*). The information extractable from such graphs is valuable to undertand the job market, hiring strategies and connectivity between industries within the same sector and cross-sector. This would benefit human resource departments, as they might gain new insights on how to improve their hiring strategies, head hunting companies, as they might find new areas where to focus their business, and individuals, who might find an useful tool to optimize their career path.

This topic has been widely treated in literature and a few example studies are used as reference. Sun B. et al [2] look at the correlation between (a) breadth and depth of firms' embedding in the talent flow network and (b) their innovative performance and life cycle. Richard J. Oentaryo et al [3] compare and constrast job networks in three areas (Singapore, Hong Kong and Switzerland), distinguishing between internal and external hops and inspecting correlations between hops and work experience, job age, number of declared skills and career evolution.

1.1 Literature Research

Nag Rajendran [1] took a snapshot of LinkedIn data and created a talent flow graph where each node in the graph is a S&P 500 firm from 2018. The edges have an attribute that is the number of employees that migrated, normalised by the workforce cardinality. The author looked at the in-degree and out-degree distribution for the unweighted graph and noted that the highest degrees are biased towards bigger companies and showed, with linear regression, that the correlation between workforce count and turnover is indeed significant. Using the weighted graph, the author notes how the biggest employees exchange happens between subsidiaries of the same company. It is also possible to see that the PageRank in the weighted case is peaked at a low value for most companies, whereas it is spread at higher values in the unweighted case: this can be interpreted like if companies with a high flow of employees, in turn had less chances to be visited by the random surfer. Finally, by using community detection and calculating assortative mixing, it is shown that employees tend to migrate between companies within the same sector.

Sun B. et al [2] use the data available from public profiles on LinkedIn China and the finantial data of China's listed companies between 2000 and 2015 to show that:

1. The breadth and depth of firms' embedding in the talent flow network positively impacts their innovative performances

2. Younger firms' innovations are mostly promoted by the breadth of the network embedding and this effect tends to fade off as the firms age

3. Mature firms' innovations are primarily driven by the depth of the network embedding

After creating a talent flow network where nodes are firms and edges are job hops, Pajek 3.0 is used to calculate degree centrality and closeness centrality, which are used to measure the embedding breadth and depth of firms in the graph. This information is combined with the Chinese Stock Market and Accounting Research (CSMAR) Database and an empirical model is created to validate the hypotheses made.

Finally, Richard J. Oentaryo et al [3] analyze the job hopping behavior to understand job preference and career progression. Job titles are important in such analysis, and a dedicated translation and parsing (called *job normalization* in the paper) is used to uniform the data collected from an OPN in three different countries. The authors analyze various aspects of a job hop, among others: whether it is internal or external to the company and how it correlates to the work experience and permanence within the same job. Also in this case in-degree centrality (talents attractivity), out-degree centrality (influential jobs, supplying talents) and PageRank (measure of global competitiveness) are used to gain insights from the network.

2 Dataset

The data for 1500 public profiles have been scraped from a professional network website; this includes company, role, start and end date for each professional's job history. Given the relatively small number of observations, the analysis focuses on the comparison between only two companies within the entirety of the website's data, as detailed in section 3. For this reason, these 1500 profiles are of individuals that have worked or work in one of these two firms. Sun B. et al [2] collected a number of employment records from China's listed companies of the order of 100000: as the number of listed companies is greater than 300 we can hypothesize an average of maximum 330 profiles per company. Richard J. Oentaryo et al [3] collected 3.5M profiles for 315K organizations; an average of 11 profiles per company. Comparing with these examples from literature, the size of the dataset collected for this study seems appropriate, considering that the focus is only on two firms (hence roughly 750 profiles per firm).

To give some context, we disclose that the two companies object of this study are in the Italian Motor Valley. However, to ensure anonymity, these are referred to as company A and company B, and so are the other companies belonging to the Italian Motor Valley nicknamed company X with $X > B$. Other companies are not nicknamed, as it seems to be common practice in literature. Employee names are not used by any means in the analysis to ensure privacy. Moreover, the raw data is transformed into a job hop database from which it would be no longer possible to reconstruct an individual's career path. It would be hard to calculate the k-anonymity if one had the career path of an individual at hand, however, given the extent of the data available online, it would be hard to think of it being less than 2-anonymous.

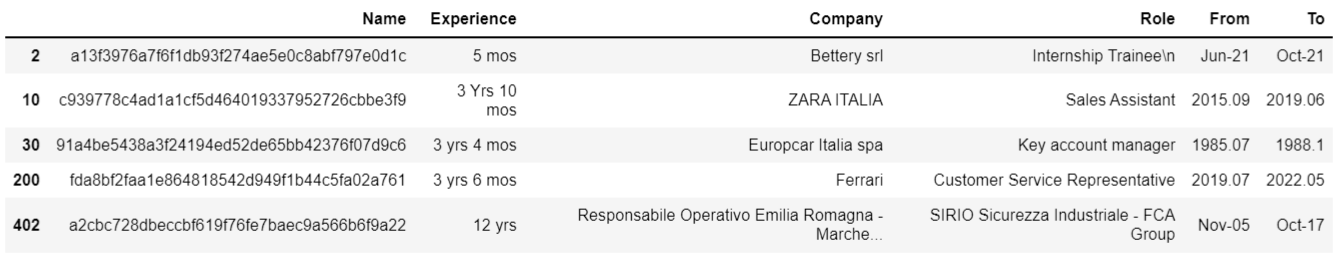
Starting from the raw dataset, the first non-trivial task is to transform it in a job-hop or role-hop dataset. In this study we focus on the job-hop network only, where nodes are companies, leaving the other for another occasion. There is the need to parse dates, company names, extract the job hops and construct the network.

2.1 Name Parsing

The raw data to replicate this part is not available, for privacy reasons. People's names have been hashed using the *SHA1* (via **hashlib**) encoding and appear as an alphanumeric strings. Figure 1 shows a few rows of the dataset.

2.2 Date Parsing

All data fields set to "present" or "Present" are substituted by the current date. Using the fact that no days are specified, but only months and years, algorithm 1 is deployed to pre-process years and ease the parsing of dates. Consequently, date parsing is done with Python's **dateparser**. Observations for which the date could not be parsed correctly or that show a job end-date prior to the start-date are eliminated.



	Name	Experience	Company	Role	From	To
2	a13f3976a7f6f1db93f274ae5e0c8abf797e0d1c	5 mos	Bettery srl	Internship Trainee\n	Jun-21	Oct-21
10	c939778c4ad1a1cf5d464019337952726cbbe3f9	3 Yrs 10 mos	ZARA ITALIA	Sales Assistant	2015.09	2019.06
30	91a4be5438a3f24194ed52de65bb42376f07d9c6	3 yrs 4 mos	Europcar Italia spa	Key account manager	1985.07	1988.1
200	fda8bf2faa1e864818542d949f1b44c5fa02a761	3 yrs 6 mos	Ferrari	Customer Service Representative	2019.07	2022.05
402	a2cbc728dbeccbf619f76fe7baec9a566b6f9a22	12 yrs	Responsabile Operativo Emilia Romagna - Marche...	SIRIO Sicurezza Industriale - FCA Group	Nov-05	Oct-17

Figure 1: Raw dataset with hashed names

Algorithm 1 Dates pre-process

```
Require: date string with only year and month
Ensure: date string where the year is 4 figures
  if there are one figure “X” or two figures “XY” preceded and followed by no numbers, i.e. isolated, and no number with 4 figures is in the string then
    if the number “X’ or “XY” smaller than current year “22” then
      “X” $\rightarrow$ “200X” or “XY” $\rightarrow$ “20XY”
    else
      “XY” $\rightarrow$ “19XY”
    end if
  end if
```

2.3 Company Name Parsing

The web scraper, due to the different positions that role and company names might take in the profiles' web page, mixed company and role names for some observations. This required a programmatic swap when necessary. The first operation is to translate the *Company* and *Role* columns to English where necessary using the Google translator module in the library **deep_translator**. Punctuation and stop-words are removed, and all words made lower-case.

A list of jobs was extracted from the website `https://www.careerbuilder.com/browse` to create a list containing the last word of each (for example, from “data engineer” we retain only “engineer”, or from “career coach” we keep only “coach”), together with some extra key-words such as “manager” or “machinist” that were not in the list; the list contains more than 50 job keywords. If any of these keywords is found in the “Company” column and not in the “Role” column, the swap takes place. Manual inspection of the database after this operation shows that only a few rows still look odd, however predominantly due to the profile containing mixed information (e.g. the role repeated in the company name).

2.4 Company Name Normalization

The roles are left aside in this study, although a natural extension would consider these too. A challenge also faced by Richard J. Oentaryo et al [3] is job normalization: unfortunately on professional network websites there is enough freedom to specify company names that the same company might be found under different names, for example “PWC” might be found as “PWC Italy” or “Price”. This in turn is a text processing task, where each firm's name is a document and the entiriety of the firms' names are the corpus. Some alternatives have been tested:

- Count vectorization of firm names and k-means clustering;
- Term-frequency inverse-document-frequency vectorization of firm names and k-means clustering;
- Doc2Vec vectorization of firm names and k-means clustering;
- Firm name's keyword extraction based on Google's n-gram term frequency;

- Matching of keyword from list of companies

It is worth noticing that the first three approaches are very general and would be applicable to the role names too. The penultimate approach would not work for role names, as “data engineer” would be transformed onto “engineer” which looses most of the information contained in the job role. The last approach would perhaps be overwhelming in the case of job roles, as there are many and with different details. Richard J. Oentaryo et al [3] used NLP to extract job type and level from job role strings.

Going back to company name normalization; in the count vectorization approach, documents are vectorized using **sklearn.feature_extraction.text.CountVectorizer** and *k*-means clustering is done with **sklearn.cluster.KMeans** with 200 clusters, random state set to 892536 and 10 initializations. This approach did not prove successful due to some words that inappropriately were included in the company name such as “fulltime”, that appeared as the discriminating term for cluster 0 membership, or “italy”, that was the discriminating factor for cluster 5 or again “industrial”, discriminating cluster 12. Fair to say that this approach did well in some cases, but it is not reliable enough.

A similar approach was attempted with term-frequency inverse-document-frequency vectorization with **sklearn.feature_extraction.text.TfidfVectorizer**. As the documents are very short, generally maximum 4 words like “eujapan centre industrial cooperation”, there was no need to set the minimum and maximum document frequency parameters. As expected, this approach does a lot better, as frequent words such as “bank” or “industrial” are weighted less. Table 1 shows some clusters that make sense. Despite the dramatic improvement over the previous method, however, some clusters like the one shown in table 2 make less sense.

The last clustering approach used **gensim.models.doc2vec**, an extension of Word2Vec by Le and Mikolov [4]. Training the algorithm on this corpus made of company names did not prove successful, as the sentences are very short and contain many atypical words and proper names. We could not find a pre-trained version, the equivalent of GloVe, but for sentences.

This though on the atypical nature of the words contained in a company name string, triggered the following experiment: extracting from each company name only the most unusual word. For example, no matter whether I have “kpmg” or “kpmg italy” or “kpmg consulting”, all these three are going to map to “kpmg” if I retain the most rare word. What comes in handy here is Google’s pre-populated n-gram available at the following link `https://books.google.com/ngrams/`. Using the library **requests**, I wrote a function to send an HTTP request to the website and return the results. The returned value is a % usage of each word in year 2019 based on the English vocabulary corpus and the least frequent word is kept as company keyword. The algorithm was able to reduce the number of companies from 2916 to 2103. This method performed quite well, however: in some cases the word “fulltime” or “spa” (Italian for Ltd.) was the least used and, unfortunately, for time constraints, the algorithm had to give up after more than 10 missed connections to the website, and that left some company names unchanged. Nevertheless, this approach looked like one of the most promising together with the TF-IDF vectorization.

The last approach used is the most “manual”: a list of 144 company names was collected, together with their workforce number and sector attributes. A simple search and substitute was done on the database. While very basic and not generalizable, this approach (a) is the most robust and controllable, (b) allows to work with a chosen smaller number of companies and (d) allows to have company attributes. It is the selected method to continue in the context of this analysis and in the given time frame.

2.5 Job-hop Network

With the data parsing and normalization completed, the final preparation step is concerned with the job hop extraction. First of all, people with only one working experience are eliminated and then job hops

Cluster	('mclaren racing','mclaren technology group')
Cluster	('federico ii university hospital', 'second university naples' 'federico ii university naples contract', 'university naples federico ii', 'unina corse squadra corse federico ii')
Cluster	('toyota material handling manufacturing italy spa toyota group', 'kyoho toyota mexico toyota group', 'attleboro toyota', 'toyota motor europe', 'radhakrishna toyota', 'alfuttaim automotive toyota workshop', 'toyota material handling italia')

Table 1: Some clusters with **TfidfVectorizer** vectorization

Cluster	('sapienza corse racing team', 'team member' 'pisa air cargo team', 'williamsf1 team', 'team srl', 'team leader', 'race team', 'team member electronic group', 'team fitness', 'ecohybrid katane team', 'salento racing team', 'racing team eteam', 'sportpesa racing point f1 team', 'ferrari f1 racing team', 'asd euro team torino', 'haas f1 team', 'team service usa'...

Table 2: A dodgy cluster with **TfidfVectorizer** vectorization

are extracted using algorithm 2 for each person in the database. The algorithm deals with parallel career paths in the case of people having parallel commitments, based on the sequentiality of start and end dates.

The list of job hops is the used to extract an unique list of job hops and a weighting factor that is given by the number of people per job hop divided by the workforce count in the source company. The normalization by workforce count in the target company is also stored. Considering the 144 selected companies, there are 954 job hops of which 454 are unique, meaning that the graph has 144 nodes and 454 edges. This was used to create directed graphs, both unweighed and weighted with the library **networkx**. Each node (company) is assigned its own sector.

The two companies subject of this study have degree 101 and 119 whereas most other companies have degree between 5 and 20 and a few have lower degree. The graph has cycles.

Algorithm 2 Job hop extraction

```
Require: database of job history ordered by start date $H = \{J_0 \cdots J_n\}$
Ensure: list of job hops where each element has the form ('source company','destination company')
  Create an empty list of parallel jobs in the queue $Q = \{J_0\}$ and set $i = 0$
  for $J_i$ in $H$ different from $J_0$ do
      for $J_j$ in $Q$ do
          if $J_i$ started after $J_j$ ended then
              Append job hop to output list
              Pop $J_j$ from $Q$
              break
          end if
          Append $J_i$ to $Q$
      end for
  end for
```

3 Targets and techniques

The target of the analysis is to extract information that could be useful to human resources professionals, head hunters or individuals. In this example we will be mainly comparing two companies "companyA" and "companyB" from the Italian Motor Valley. This is the list of some of the questions we would like to

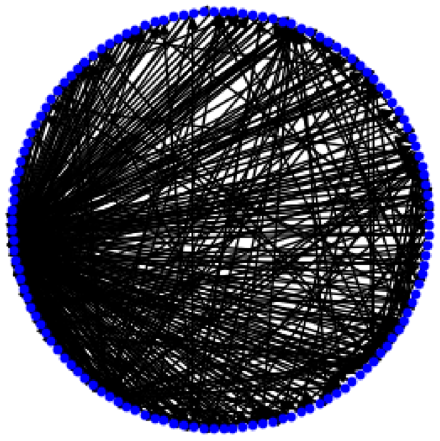


Figure 2: Job hops graph

answer:

- What does the turnover look like for the two companies and how does it compare? Using the terminology of Richard J. Oentaryo et al [3], which company is more prominent (attracting talent) and which one is more influential (giving talent)?
- Between what companies is there the greatest flow of talent? And in which direction?
- How deep and how strong do the companies subject of this study look for talent? Using the thesis of Sun B. et al [2] this would be related to asking: which company is likely the more innovative and/or the one at a later life stage?
- At which company talents aspire to work the most? How competitive is the company in the job market?
- Based on the data available, what would be the shortest path for an employee to get from a company to another?
- Can we detect any communities? Are these clustered by sector? To which communities do the companies object of this investigation belong?

This are all questions that could be answered with the methodologies explained in the following sections.

3.1 Turnover: in-degree and out-degree

Understanding the talent flow through a company in turn means looking at the number of incoming and outgoing edges in a job network. For the unweighed graph we will look at in-degree, out-degree, but also in-degree-centrality and out-degree-centrality which are the fraction of nodes that are connected to the node in question. For the weighed graph it is also possible to calculate the weighed version of in-degree and out-degree. Note that for the in-degree it makes sense to use the weighting based on the workforce count of the target company, as knowing which percentage of the source company's workforce is incoming is not important. Richard J. Oentaryo et al [3] assert that the higher the in-degree-centrality the more prominent (attracting talent) the company is, the higher the out-degree-centrality the more influential (giving talent) the company is.

3.2 Where it is happening: sub-graph with companies with highest weights

Understanding a bigger picture of which edges have the strongest connections is in turn a matter of analysing a sub-graph of the original one, where only edges above a certain weight threshold are kept (for completeness, in this case we are considering the weighting based on source company workforce).

3.3 Innovation and age: degree and closeness centrality

This point follows Sun B. et al [2], who look at degree and closeness centrality. Degree centrality is the fraction of nodes that connect to the node in question, closeness centrality is the reciprocal of the average shortest path distance from the node in question to all reachable nodes. Low values of closeness centrality mean that most connected nodes are not directly reachable, whereas high values mean that most nodes are directly connected via a 1-edge path. The higher the degree centrality, the more connected the firm is (breadth in the network): the authors link this to a start-up or young company approach where talent is needed from different areas to innovate. The lower the closeness centrality, the fastest/strongest the connection is: the authors link this to older companies that have built stronger links as need to perfect their product. Both breadth and depth in the network contribute positively to the company's performance, based on the analysis in Sun B. et al [2].

3.4 Competitiveness: Google PageRank

This problem can be addressed using PageRank, an algorithm developed by Google to rank web pages in order of importance. A firm is considered important if many other firms "point" to it and if they are important too, meaning that they have many out-links too. Let $L_{ij} = w_{ij}$ if firms i and j are connected and zero otherwise, and $c_j = \sum_{i=1}^{N} L_{ij}$ be the number of firms pointed to by firm j, then the page ranks are defined as

$$p_i = (1-d) + d \sum_{j=1}^{N} (L_{ij}/c_j)\, p_j$$

where d is a positive constant. The importance of firm i is the sum of the importances of firms pointing to it, weighted by $1/c_j$. The constant d is to ensure that each page gets at least importance $1-d$. This metric can be used to calculate how competitive a company is in the job market with respect to others. In our case we set the parameter $d = 0.15$ and calculate both the weighted and unweighed versions. By looking at the distribution of page ranks between these two versions it is possible to deduct on the effect of the weights. As we weight for the size of the source company, w_{ij} becomes smaller for bigger companies, and having average higher page rank would mean that small companies are generally more influential. The vice-versa also holds.

3.5 Fast track to a new job: shortest path

The problem of the fastest track between two companies can be solved by finding the shortest path (Dijkstra's algorithm in the weighed case). The weighed graph gives insights considering the percentage of workforce that leaves the source company in a given direction: this adds useful information on which path is also more "likely" other than faster.

Calculating all shortest paths from a given company to all others, or the shortest paths to a company from all others, it is possible to gain a more complete overview of the job hopping phenomenon. These were calculated using the functions **single_source_shortest_path_length**, **single_target_shortest_path_length** for unweighed graphs and the equivalent functions for weighed graphs. Note that the function **single_target_dijkstra_path_length** is not in the package **networkx**, hence was created based on **networkx.dijkstra_path** as explained in algorithm 3.

Algorithm 3 Dijkastra Single Target Shortest Path

```
Require: directed graph $G$ and a target node $j$
Ensure: list of shortest paths from any node $i$ to $j$
  Create an empty dictionary of path lengths with key $i$
  for $i$ in $G.nodes()$ do
    if there is a path connecting $i$ with $j$ then
      $l_i = networkx.dijkstra\_path(G, source = i, target = j)$
      update the dictionary with key $i$ and value $l_i$
    end if
  end for
```

Company	A	B
In-degree	81	103
Out-degree	21	16
In-degree centrality	0.623	0.792
Out-degree centrality	0.154	0.123
Weighed graph in-degree	0.171	0.069
Weighed graph out-degree	0.044	0.061

Table 3: In-degree and out-degree

3.6 Linked companies: community detection

Community detection is done using the Girvan Newman algorithm, iterating until the biggest cluster has maximum 10 firms. All communities with less than two firms are then set apart.

Another measure that helps at understanding how similar firms connect to each other are is assortativity, which could be based on many similarity measures. One of the most common in network analysis is degree assortativity; the coefficient is the Person correlation coefficient of degree between pairs of linked nodes. It is possible to calculate correlations between out-degrees, in-degrees or mixed out-in and in-out degrees, both for the weighed and non-weighed graphs. The Pearson correlation can be calculated for any node attribute and, in this study, we calculate it for the company's sector and compare it to the clustering results.

4 Results

4.1 Turnover: in-degree and out-degree

The in-degree and out-degree for companies A and B is shown in table 3. From the relative difference between in-degree and out-degree, it is possible to notice that company B is hiring from more companies than it is giving to and, indeed, has a higher in-degree centrality and lower out-degree centrality than company A.

By looking at the weighed graph (where each edge is weighed by the number of people that moved between companies divided by the workforce count in the target company) it is possible to see that company A is hiring more than B from the companies it is linked to and fewer people are leaving, in relative terms.

In conclusion, from the unweighed graph we see that company B is more prominent (attracting talent) than A and less influential (giving talent). From the weighed graph we also see that company A is hiring more than B through the links and giving away less.

Company	A	B
Degree centrality	0.78	0.92
Closeness centrality	0.72	0.82

Table 4: Degree and closeness centrality

4.2 Where it is happening: sub-graph with companies with highest weights

The top 20 links from the weighed graph are selected and shown in figure 3. Company A and B are unsurprisingly strongly linked and so they are with other companies in the Italian Motor Valley such as C and E. Company A is more strongly linked to Mercedes, McLaren, HT, Volkswagen than company B, which is more strongly linked to Porsche, BMW, Accenture and Bosch.

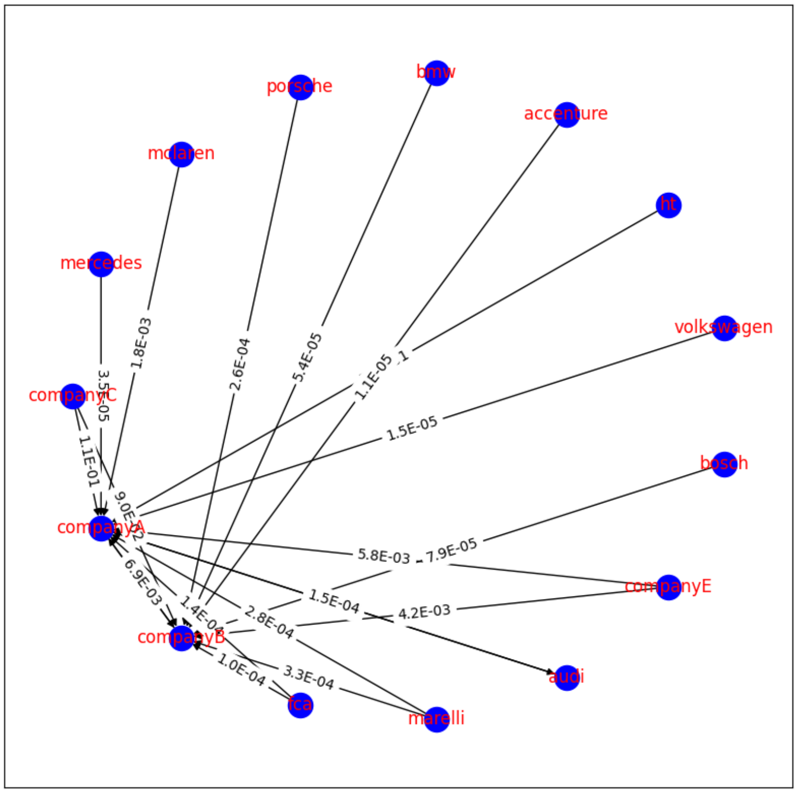


Figure 3: Firms connected by the heaviest links in the weighed graph

4.3 Innovation and age: degree and closeness centrality

Table 4 shows degree and closeness centrality for the two companies. Company B has a higher degree centrality, hence it is connected with more firms in the network. In terms of closeness centrality, company B is looking less far in the network. These indices would be in favour of hypothesizing that company A is slightly less innovative but looking for or attracting specific talents through the network.

4.4 Competitiveness: Google PageRank

The page ranks for companies A and B are shown in table 5. Company B is more competitive in the job market than company A in both cases.

Company	A	B
Unweighed graph page rank	0.039	0.051
Weighed graph page rank	0.050	0.061

Table 5: Google Page Rank

4.5 Fast track to a new job: shortest path

Let us show the power of the network in understanding the shortest path between companies by a simple example: an employee working in automotive, either in company A or B, dreams to move to the fashion industry working for Bulgari. In the network there is not a direct path between either company A or B to Bulgari, indeed the following shortest paths are found:

$$companyA \rightarrow companyE \rightarrow Bulgari$$

$$companyB \rightarrow companyA \rightarrow companyE \rightarrow Bulgari$$

Working at Bulgari in both cases requires passing through company E in the motor valley, which is connected only to A, hence from B a pass through A is needed. Considering the weighed paths the situation is the following:

$$companyA \rightarrow companyE \rightarrow Bulgari$$

$$companyB \rightarrow M.Marelli \rightarrow SegulaTech. \rightarrow R.Rover \rightarrow companyE \rightarrow Bulgari$$

Interestingly, the path stayed the same for company A, whereas the link between B and A was so weak that an easier weighed path reaches company E via Magneti Marelli, Segula Technology and Range Rover.

By calculating shortest paths for all nodes, both as source or target, it is possible to see the easiest or most difficult firms to reach from A and B or from which companies it is most difficult to get to A and B, if there is a connection. Here we exemplify the going to A and B. In the unweighed graph we find that it is most difficult to reach company A with 2-step path from many companies such as Barilla, Siemens, Azimut and others. For company B, Chrysler employees need a 3-steps path to get to it.

Considering the weighed graph, it is difficult to reach Company A from Alpinestars and HT (5-steps path), whereas Iveco and Chrysler are the firms from wich it is hardest to reach company B (5-steps path).

4.6 Linked companies: community detection

After running the community detection algorithm until the biggest cluster has 10 elements and removing the smallest clusters with up to 2 elements, we found ourselves with 10 clusters: (‘Alma Automotive’, ‘Bulgari’, ‘companyA’, ‘companyB’, ‘companyC’, ‘companyE’, ‘Mechinno’, ‘Piaggio’, ‘Pirelli’, ‘Land Rover’), (‘Altran’, ‘Ansaldo’, ‘CNH Industrial’, ‘FCA’, ‘FIAT’, ‘Italdesign’, ‘Turin Polytechnic’), (‘Alstom’, ‘Marelli’, ‘Siemens’), (‘Almani Group’, ‘BNP’, ‘Decathlon’), (‘Audi’, ‘Lotus’, ‘Mercedes’, ‘Porsche’, ‘Renault’, ‘Toyota’, ‘Volkswagen’), (‘FEV’, ‘FPT’, ‘Valeo’), (‘Bottega Veneta’, ‘Chrysler’, ‘Hugo Boss’), (‘CRIF’, ‘KPMG’, ‘Louis Vuitton’), (‘AlphaTauri’, ‘Toro Rosso’, ‘Williams’), (‘Rolls Royce’, ‘Seat’, ‘Tesla’).

The first cluster contains most companies from the Italian Motor Valley and from the Emilia Romagna region such as Alma Automotive. All companies in cluster 1 except Bulgari are in the automotive sector. The second cluster makes sense too, as it is the Turin-centric cluster: it mostly contains companies and universities head-quartered in Turin or Genoa. So far we have found the Italian north-east and north-west clusters. Without going further in detail, as that would require a specific knowledge, we note here that we find other meaningful clusters such as the central-european automotive cluster (‘Audi’, ‘Mercedes’, ‘Porsche’, etc...), the Formula 1 cluster with (‘AlphaTauri’, ‘Toro Rosso’, ‘Williams’) and a financial cluster with ‘KPMG’ and ‘CRIF’.

Finally, the assortativity coefficient is calculated to answer questions like: how similar is the in-degree or out-degree of connected companies? How similar is the sector of connected companies? Table 6 shows

Metric	Value
In-degree assortativity	-0.21
Out-degree assortativity	-0.23
In-degree assortativity weighed graph	-0.209
Out-degree assortativity weighed graph	0.003
Company sector assortativity	0.033

Table 6: Assortativity coefficients

the resulting coefficients. The only positive correlation is seen in sector assortativity (automotive, fashion, academic,...), and this lines up with the clustering shown before.

5 Conclusions

Using data from public profiles in an online professional website, a job-hopping network is created. Network analysis techniques are used to extract useful insights from the dataset, and their effectiveness is exemplified using two companies in the Italian Motor Valley nicknamed as A and B. The results of the analysis make sense, which increases trust in the database and techniques used.

The natural extensions of this work are:

- Persist with the automatic job normalization with TF-IDF vectorization and clustering or by means of Google's n-gram, allowing to extend the final network to more than 144 nodes
- Use an external dataset for firms' meta data to look at correlations between their positioning in the job netork and other attributes
- Extend to the job role network as the raw data is already available

6 Code

The code used for this study is available on GitHub at the following link `https://github.com/a-mazzetto/job-hop-net`. This code was run on a DELL laptop XPS 13 7390 with processor Intel(R) Core(TM) i7-10710U CPU @ 1.10GHz 1.61 GHz, RAM 8.00 GB (7.79 GB usable). Running times are quite high for the data preparation part, in particular Google Translation and Google n-gram data extraction (also affected by stale requests). The raw dataset is currently unavailable to the public, as it is still relative small and the k-nonymity has not been double-checked. Please get in touch if you are interested in using this dataset for any of your research. The post-processed dataset used to construct the graph(s) is publicly available.

References

[1] Nag Rajendran. (2021) *LinkedIn Talent Flow Network Analysis.* `https://rpubs.com/nr2462/845876` [Accessed 11th Deceber 2022]

[2] Sun B, Ruan A, Peng B and Lu W. (2022) *Talent Flow Network, the Life Cycle of Firms, and Their Innovations.* Front. Psychol. 13:788515. doi: 10.3389/fpsyg.2022.788515

[3] Richard J. Oentaryo, Ee-Peng Lim, Xavier Jayaraj, Siddarth Ashok, Philips Kokoh Prasetyo, Koon Han Ong, Zi Quan Lau. (2018) *Talent Flow Analytics in Online Professional Network.* Data Science and Engineering (2018) 3:199–220 `https://doi.org/10.1007/s41019-018-0070-8`

[4] Quoc Le and Thomas Mikolov (2014). *Distributed Representations of Sentences and Documents.* `http://arxiv.org/abs/1405.4053v2`