Skills Job Advisor

Btara Truhandarien
CMPT 733
btruhand@sfu.ca

Bhuvan Chopra
CMPT 733
bchopra@sfu.ca

Grace Kingsly
CMPT 733
gkumares@sfu.ca

Mohammad Wali Ullah
CMPT 733
mullah@sfu.ca

Problem Statement

There is no shortage of queries on how a person can be more employable in an age and era of job diversity, but also fierce competition, and as more people consider moving between jobs more frequently. We believe that a key factor in making oneself more employable for a particular job is the number of relevant skills that person has. However, in order to cultivate those skills, they must first know what the relevant skills are. In this project we attempt to answer the question of “what skills are relevant for a job?”, an exploratory work, and ultimately be able to give automatic advice to a person on the skills that they need to cultivate in a data-driven approach, a modeling work.

Motivation

As stated in the abstract, an article mentions that job-hopping, the act of moving between jobs in relatively short amount of time, is on the rise. Around 64% of workers from a survey responded they are in favour of job-hopping, “up 22% from a similar survey four years ago”. The reare 1.25 billion query hits on Google for the query “how to be a cook”, and 280 million hits for “how to be a software engineer”. There is clear interest on knowing how to work in a particular job and we believe that it is beneficial for individuals if we can give an automated, data-oriented approach to answer these questions (or some part of the question, such as the skills that they need).

We also believe that it is useful for various forms of organizations to know the relevant skills for a job. Such organizations include but not limited to universities and EdTech companies such as Coursera and Udemy that are trying to create relevant course programs for their students. By knowing more precisely the skills required for particular jobs they can create better, more tailored courses for students. Similarly, any company that is serious about training their workforce would likely benefit from this kind of endeavor, as then their HR department can create a suitable training program by comparing the skills of their workforce with the desired skills in the upcoming future.

Background

To our knowledge there is no exactly equivalent work prior to ours on this topic. However, one may view an advice as simply a recommendation and there has been many work on recommendation systems. Still again, our work is rather different from the usual recommendation scenario where a user is recommended a particular item e.g video, topic based on some preferences.

In our problem we have the following scenario

- Let *P* be an entity with some skill-set *S*
- *P* has a desired target job *T*
- There is a set of skills *R* that are relevant for *T*
- Come up with an advice *A* such that *P* is fit (or categorizable) for the job *T* given its base skill-set *S*

The advice *A* can itself be seen in multiple ways, but for the scope of this project we define it as he **difference between the skillset of *S'* and *S***, where *S'* is the skill-set of *P* when it is fit for the job.

This is where our project deviates from the usual recommendation system. Instead of recommending a job for the entity (or hence to classify), we want to modify the entity towards a particular desired state. However we do borrow the idea of using closeness to a neighbouring data point in a recommendation system.

The idea is that within a cloud of skill-set each clustered to a group of jobs (or hence assigned a job title), the skill-set *S* of *P* can be put in this cloud. If *T* is the target job with a cluster of skill-sets *C* that are suitable for *T*, then we can try to find a skill-set S_C in *C* and find the **difference** between *S* and S_C, which by our definition is a suitable advice. A possibility of S_C can be the closest point closest to *S* by some measure of distance.

Dataset collection

Our primary data source comes from Indeed's resume repository (resumes.indeed.com). We chose to use resumes because we want to capture the actual real-life job market. Instead of using job descriptions which are often idealized versions of an employer, resumes should better reflect what kind of people work for a particular job. Additionally, each of these <person, job> pair can form a single skill-set and there are many jobs that a single resume can list, increasing the number of data points we have.

To get the data from Indeed we had to crawl through the website by first initiating a query. Our query format was the following:

- A job title
- Location: Canada
- Limit search to: Job title
- Last updated: show all resumes

We wanted to cover a variety of jobs and cover a variety of industries since if we focus on too similar jobs, whether by role or industry, then it will not be as interesting and the impact of the project would be too narrow or possibly none at all (since no new insights may be gotten). The job titles we chose to search for was initially curated by taking the job titles listed in the O*NET database, an American occupation information database. We chose job titles covering a range of industries that yielded the highest result from an initial search on Indeed. The final list is:

a) **Cashier**	b) **Cook**	c) **Data Engineer**
d) **Data Scientist**	e) **Financial Manager**	f) **Hair Stylist**
g) **Host**	h) **Janitor**	i) **Office Clerk**
j) **Payroll Clerk**	k) **Production Clerk**	l) **Sales Representative**
m) **Secretary**	n) **Software Engineer**	o) **Waiter**

From Indeed resumes we took the following information:

- Skills
- Job titles, company and details
- Education (unused in the end)
- Summary
- Additional information

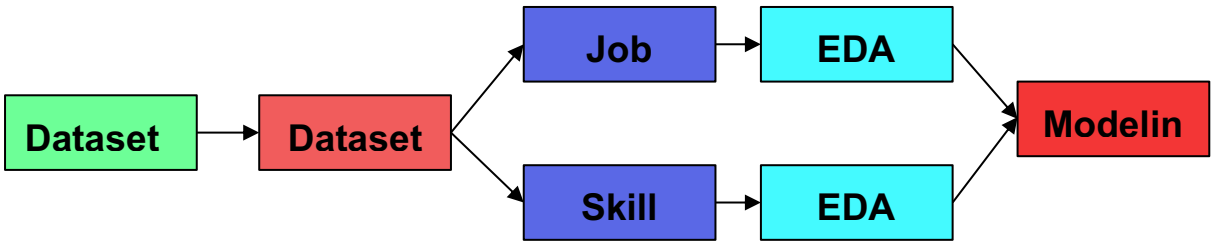
In total we got **117,712** resumes, and roughly around **470,000** job information from the resumes. We developed the scraper tool using Python, BeautifulSoup and Selenium and it can be found in this repository: https://github.com/btruhand/indeed-resume-scraper.

Strategy to tackle the problem

Since we do not have ground truth labels, our strategy is immediately to treat this problem as a an unsupervised learning task. The unsupervised learning methodology we chose was K-Means clustering due to its simplicity to implement. In order to build our envisioned model we identified two major requirements: **skill extraction from text** and **job title normalization**.

Though we got the skills information section from the resumes, people do not always put their skills in the skills section, and even if they do it may be incomplete i.e skills left out because they think it isn't important to list. Those skills may be hidden from their job details, summary or additional information.

The way that people name job titles can vary wildly and often can be very specific to what they did in their job e.g java software developer as opposed to just software developer. Since we would like to be able to assign meaning to the clusters produced, it is desirable for us to have a set of common job titles. In order to do the above we followed the following data pipeline steps:



Dataset Cleaning

Our initial dataset cleaning procedure consisted of two parts:

1. Format dataset suitable for future ingestion
2. Tokenizing summary, additional information and job details

Dataset formatting was done to ease future work. Mainly we simply transformed each resumes to JSON objects holding information in lists of strings or as paired information such as triplets of job title, duration and details. A special note on duration of jobs and study, Indeed allows the date of study or job to be "Present", indicating the person is still studying

or working at that job. However, we treat these values as NaN values because we don't have a way to actually know if they are truly still at present studying or working and we simply opted to treat them as not having set a time duration at all.

Summary, additional information and job details are all text blobs. Often they will contain bullet points which we simply treat as another sentence. To tokenize these text blobs we took advantage of NLTK tokenizer and used its *sent_tokenize* and *word_tokenize* function to tokenize a text blob into sentences and sentences into individual words. We also removed common english stopwords as provided by NLTK's stopwords from its English corpus. We also made every word to be lowercase to reduce variability.

Job Title Normalization

Entity Resolution with Jaccard Similarity

Our first attempt at job title normalization is to take advantage of O*NET's alternate titles information. They provide a list of job title and alternative titles pairs that have been used to refer to the that particular job title. From our cleaned dataset we expanded all of the available jobs from the resumes we have gathered so that each resume may give more than one job (note that it may even give 0 jobs). Since some people have opt out not to include their job title we remove those data points. Next we tokenize the job titles and alternative titles (a simple tokenization based on whitespace). We use those tokens to perform entity resolution between using Jaccard's similarity as was done in one of the assignments. Our hypothesis was that using Jaccard similarity we hope to be able to identify job titles that has a high enough jaccard similarity with some alternative title and use the job title from O*NET as the normalized title.

We find that the results were not too bad, but also not good enough for our needs. Consider the following pie chart

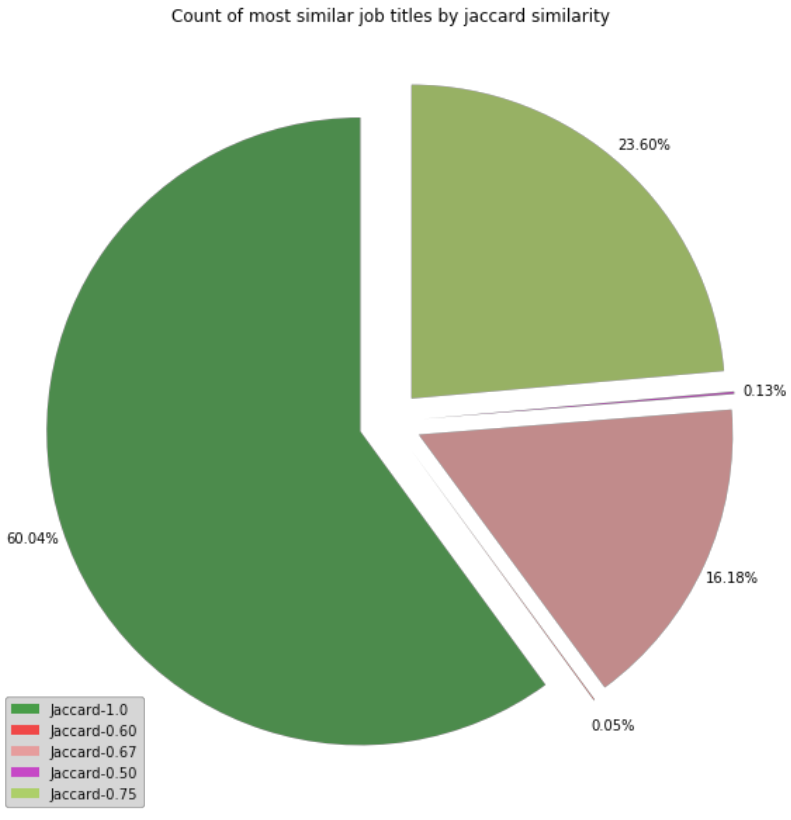


Figure 1 A pie chart showing the proportion of Jaccard Similarity values amongst all of the highest Jaccard similarity for each job title

We have a list of jaccard <resume job title, alternative title> pairs along with their jaccard similarity. For each

resume job title (identified by an assigned job ID we created), we find the highest Jaccard similarity that managed to be computed with a threshold of greater than 0.5, creating a list of most similar pairs. There are in total over 390,000 of them. The above diagram tells us the portion of each Jaccard value in this list of most similar pairs.

Getting 60% of the result to have a Jaccard similarity of 1 (perfect similarity) is a good thing. However the number of unique normalized job titles we got is **531**. Moreover the distribution of the job titles are is not even, implied from the following statistics:

Statistics category	**Count of normalized job title that got matched**
Maximum	7527
Minimum	1
Median	15
Average	173

For the dataset that we gathered we felt the number of normalized job title is too high and each normalized job title in itself is only matched with a few resume job titles.

Moreover, we notice that inspecting the other Jaccard similarities we get a few strange results:

Alternate Title	Resume Job Title	Jaccard Similarity
Vice President Talent Management	Vice President Product Management	0.6
Wind Operations Manager	Assistant Operations Manager	0.5
Software Applications Specialist	Software Applications Engineer	0.5

Some of the results make sense to be taken as similar jobs, at other times it is unclear just by inspecting the Jaccard Similarity. Clearly it is better if we can consider the context of the job details too

Doc2Vec methodology

We then tried to move over to a Doc2Vec solution. Again, we took advantage of available information from O*NET Database. There are two kinds of documents we thought to be suitable to use from the O*NET Database, information about job descriptions and job task statements.

A job detail in a resume we thought can be seen as either a description or a list of tasks during the time of the job. Our idea is then to combine the job descriptions or task statements along with it the job details from all the resumes and treat it as a document corpus. Then we hope to use Gensim's implementation of Doc2Vec in order to get document embeddings for all of the documents in the corpus. Then we hoped to be able to find the job

description or task statement document from O*NET that is most similar for every job detail document.
For this part our cleaning process was a bit different and we used SpaCy's tokenization, due to its supposed ability to retain as much of the original text as possible so as to not lose information. Stopwords were not removed due to its overall effect on Doc2Vec and lastly we modified DATE, TIME, ORG, and GPE entities into standard tokens to minimize variability that should not be taken into account.

What we found was that overall results were not good. The following are box plots of the similarity of job details against either job descriptions or task statements

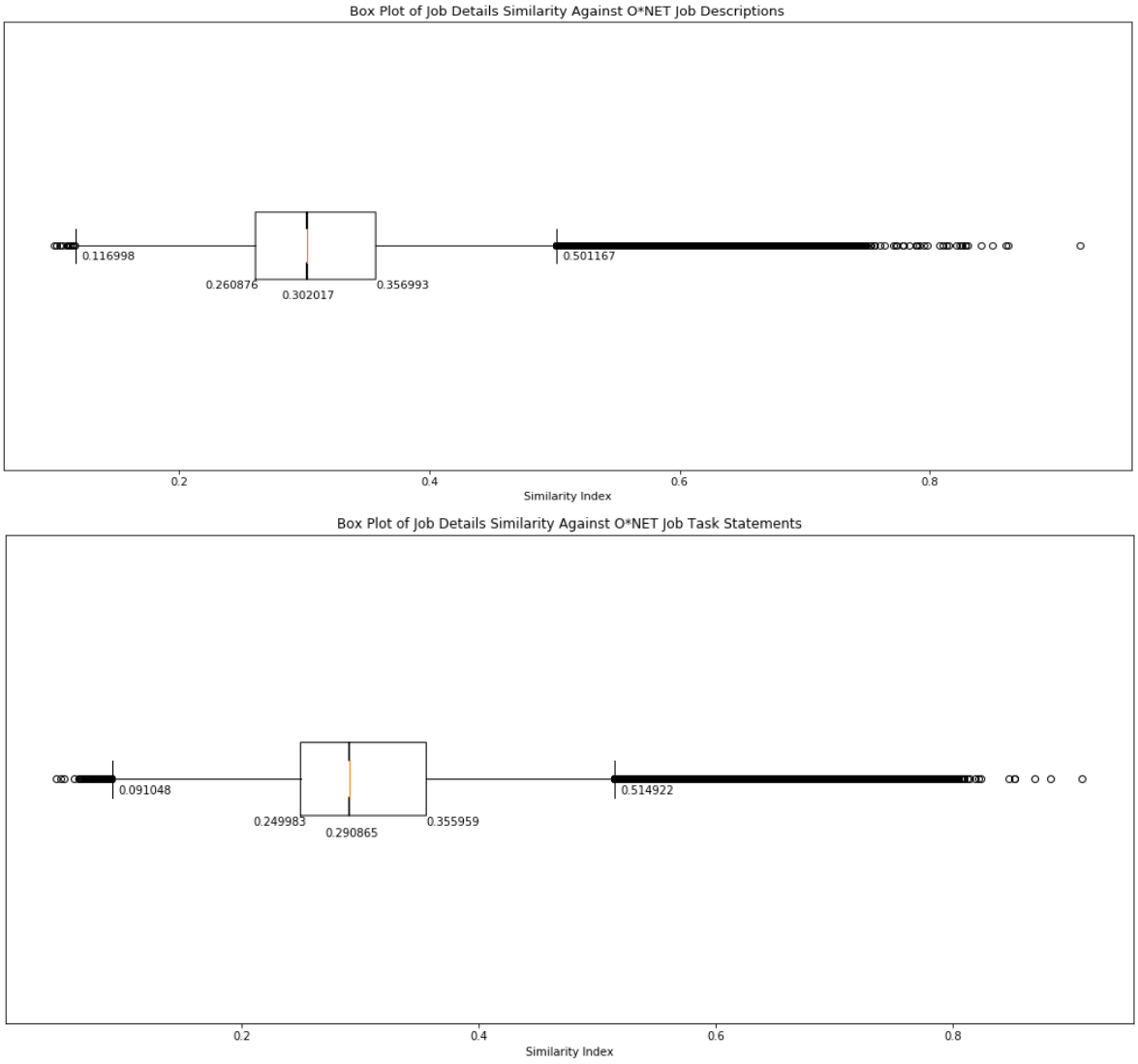


Figure 2 Box plots of similarity (by cosine similarity) of job details document embedding on O*NET document embeddings with the highest similarity. Higher is better

The box plots ultimately tell us that for 75% of the job details, the O*NET document that they are most similar with only has a similarity value of around 0.5 at most. Ideally the median of similarity should be higher (shifted to the right). We didn't dwell much on why this happened, but some possible factors may related to preprocessing, or just inherent differences in the qualities of the job detail documents and the O*NET documents, such as grammar or

general focus in content. We note that the Doc2Vec training itself is fine, yielding 93% of the documents to be rated most similar to itself.

Skill Extraction

Though we weren't able to come up with a good way to normalize job titles, skill extraction is absolutely essential if we hope to get the people's skill-sets from the resumes. People generally like to put some sentences in their resumes such as "Developed an application using X, Y and Z" in their work experience details. In our data, such kind of sentences were there in columns named 'summary', 'additional', and 'job details'. In order to get the skills from unstructured text, we calculated the Term Frequency - Inverse Document Frequency (**TF-IDF**) value on a corpus of text generated from these columns.

Building Vocabulary

To do this, we needed to build a vocabulary of the skills that people have. This vocabulary was built using the data from the O*NET database as well as the skill column from the resumes data. We downloaded occupational information such as knowledge and ability required for a particular job, and information like general and technical skills that were necessary to do that job from O*NET. We also got skills that people have put in their resumes in the skill column as mentioned before. In skill column sometimes, skills are well structured comma separated words. In contrast, in many case, they contain long sentences with bullet points and other usual formatting. Hence, we have done similar cleaning as mentioned in the "Dataset Cleaning" section. Finally, the combined vocabulary from O*NET and skill column contained 42761 unique skills.

Calculating TF-IDF scores

After we got our vocabulary and preprocessed our corpus (removed stop-words, lowercase), we used *CountVectorizer* and *TfidfTransformer* from *sklearn.feature_extraction* library to get the TF-IDF matrix, treating the resumes as a bag of words. We set the n-gram range from 1 to 3 in order to get skills with multiple words as well. Finally, we got a dictionary of top 10 skills as the keys and corresponding TF-IDF scores as values. The following word cloud shows the common skills required for a Data Scientist:

Figure 3 Word cloud of extracted skills from Data Scientist resumes. Bigger text refers to higher relevance based on the TF-IDF score

As can be seen from the small example above, the results are a mix of skills we would expect, and some that we don't immediately associate with Data Scientist jobs. One possibility for this is because in a resume there are multiple jobs, and we simply took the

resume as a whole and did not differentiate between jobs since we could not get a good job normalization result.

Modeling

Since we could not complete our job normalization task as we desired, we weren't able to build the initial model we had in mind. Therefore we moved over to implementing a different model but still using clustering. We tried to build two models, one with Word2Vec and the other Doc2Vec. They are used to obtain vector embeddings of the skills and job titles that can be clustered. Again we used Gensim's implementation of Word2Vec and Doc2Vec and for clustering NLTK's clustering package

Clustering with Word2Vec

Our goal is two folded. Primarily, the model should cluster similar skills in close proximity and secondly, a job title should also produce relevant skills. We combined all resumes in one Dataframe and kept only job titles and skill information from the resumes. All skills were cleaned and tokenized and the job titles lowercased and tokenized. Each resume produces a list of a set of skill tokens (may be empty). The second goal requires to have the job titles also in the embedded vector space. To achieve this, we put relevant job titles in random positions in the skill list. After that, we used Word2Vec, trained using the *continuous bag-of-words* algorithm, to get a list of vocabulary and word embedding vector. Here is a example of top 10 similar/relevant skills (cosine similarity) close to *data science* after Word2Vec embedding,

Job of interest	Relevant skills	Cosine similarity
Data Science	Regression Analysis	0.847376823425293
	Natural Language Processing	0.8392703533172607
	Visualization	0.8285447359085083
	big data analytics	0.8191854953765869
	Collaborative Filtering	0.8164248466491699
	Data Wrangling	0.814964771270752
	Logistic Regression	0.809734582901001
	Experimental Design	0.7947606444358826
	Adobe Analytics	0.7931075096130371
	Machine Learning	0.7917285561561584

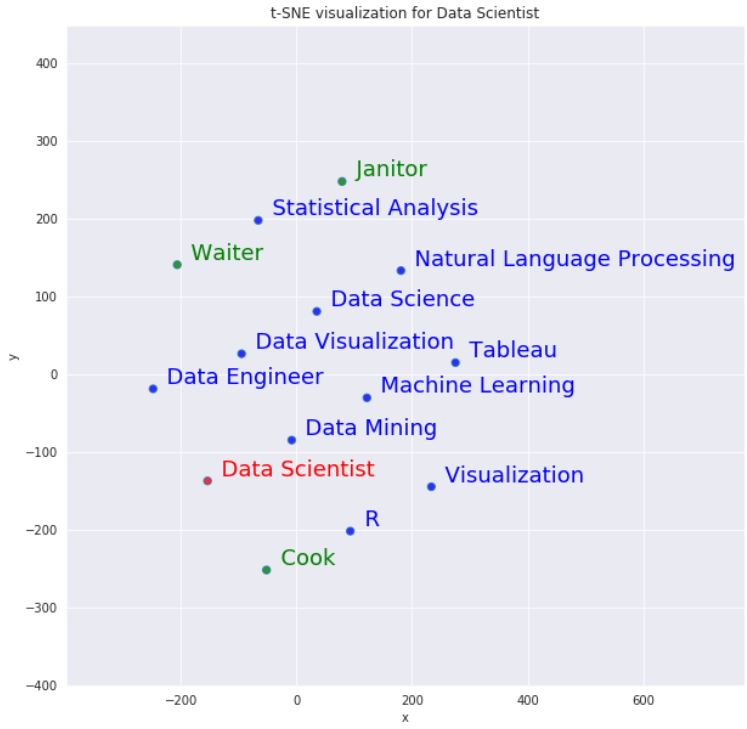


Figure 4 t-SNE scatter plot of a target job title (Data Scientist-Red), top 10 relevant skills (Blue) and comparitive jobs (Green). Results gotten from one of the earlier attempts

From the above figure we can see that we were able to retrieve very related skill-sets related to data science. For comparison purposes we just show other job titles and where they are positioned in the 2D plain. Notice that there is disparity between the visualization and what we're trying to do. The visualization tells us that the green points are not far away euclidean distance wise to the skills, but we are comparing against similarity.

After this, the word embeddings were given as input for the K-Mean clustering algorithm. We have chosen number of cluster as 15 (the number of different job titles). The word embeddings for the jobs were given as the initial mean of the K-mean cluster. The following figure shows the 10 skills which are in same cluster as job title "Secretary" and they are the closest skills from "Secretary" calculated using reduced coordinate from t-SNE.

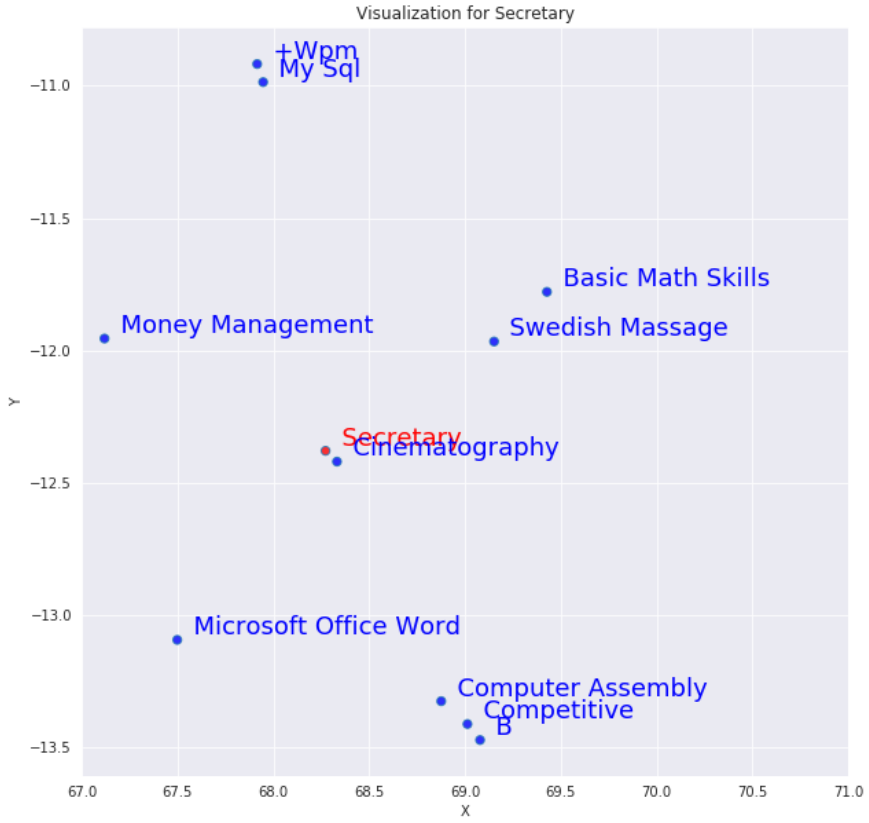


Figure 5 10 skills (blue) in the cluster of "Secretary" after K-Means clustering. Based on initial results

The Doc2Vec-based clustering model follows the same general idea. However instead of putting the job titles as part of the skills token, we let the job titles be the documents we want vectorize. To train the model we used the Distributed Memory (*PV-DM*) algorithm instead of *CBOW*. At the same time we trained the models to also train the word embeddings simultaneously along with the document (job title) vectors with the skip-gram algorithm. From our quick research this may allow the two kinds of embeddings to be more relatable. This setup trains the job titles to be predictors of the skills and we hoped that we can cluster them to the same vector space through cosine similarity.

For both models, results can be further seen in **Appendix**.

Using either of the model, we can then grab the skills within some target job cluster and compare it to a given skill-set. The missing is then the "advice" on what skills need to be cultivated.

Thoughts and Evaluation

Due to the nature of an unsupervised learning, we found it challenging to assess our model, and in general we don't have a good way to quantify what makes a good advice. Here we lay out some of our thoughts and basic evaluation on the model.

One of the first important rough evaluation is how the job titles are clustered. The reason why we chose *k=15* clusters for K-Means is because ideally there is a clear separation between each job title such that each land in their own cluster. The following is a table of the clusters of each job title (comparisons should be made against the same column)

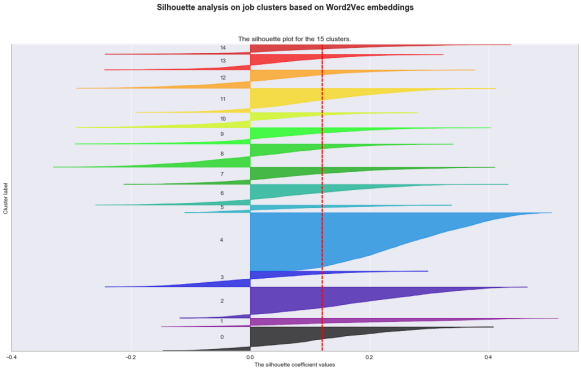
Job title	Assigned cluster using Word2Vec embeddings	Assigned cluster using Doc2Vec embeddings
Software Engineer	8	12
Host	13	14
Secretary	8	12
Payroll Clerk	4	12
Cook	11	14
Office Clerk	12	2
Financial Manager	4	6
Hair Stylist	9	14
Data Engineer	6	8
Janitor	4	14
Waiter	2	14
Data Scientist	12	11
Sales Representative	12	12
Production Clerk	4	13
Cashier	11	14

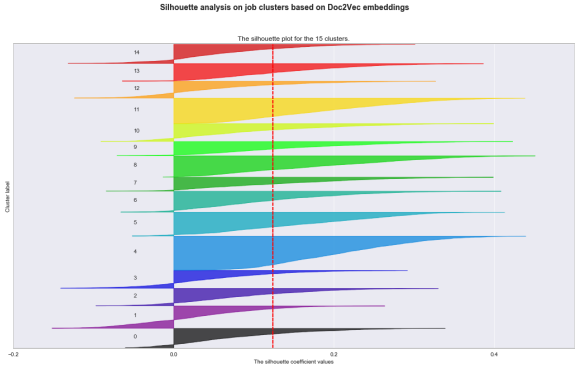
In either of the models we can see that a single cluster may have more than one job title, so immediately we know our ideal scenario is not realized. At first glance the number of unique clusters assigned to job titles is more when using Word2Vec embeddings than it is for the Doc2Vec embeddings. However, if we were to check **Appendix B** and **D**, it seems that the Doc2Vec version better captures actual related skills to a particular job. Both methodologies are not perfect and there are clear skills clustered together into the same cluster that should not be, or only vaguely related.

For a more qualitative approach we did a simple silhouette analysis of the result. The silhouette analysis gives a measure of how good the clustering is by computing how similar every point is to other points within its cluster, cohesiveness, and how further apart are each point to every other point not within its cluster, separation.

For the model with based on Word2Vec embeddings, the silhouette score calculated by SciKit learn function is **0.120621**, and the one based on Doc2Vec gives a silhouette score of **0.123983**. The higher the silhouette score, the better it is.

The following two plots are silhouette plot after doing silhouette analysis on the two models





Seeing the silhouette plots, we can see how uneven the thickness of the clusters is (indicating size of the cluster). We also see there are more data points that receive a negative silhouette score in the Word2Vec based model. This indicates that more vector embeddings from the Word2Vec-based model are poorly clustered than the Doc2Vec. From this simple analysis we can say that our approach with Doc2Vec yielded a better (or more coherent) model with respect to clustering.

These models are clearly far from perfect. One big disadvantage is that it is unable to capture the subtleties of skills and how they are actually related to many possible jobs. Skills like writing and communication are such examples. Due to a lack of time we have stopped our efforts here.

Summary

In this project we have demonstrated various techniques that we used to tackle the challenging problem of giving advice on the matter of skills that one should cultivate to be more suitable for a particular job. Our data reflects real world scenarios and people by using information from resumes, gathered by searching for resumes of people that have worked in one of 15 different jobs. With information retrieval techniques such as TF-IDF we are able to build a corpus of relatable skills for a particular job. Attempts at job normalization, required as part of our initial modeling plan, yielded unsatisfactory results, ultimately leading us to another kind of model. The final model that we built is a K-Means clustering model, with the supplied data points being document and word embeddings of job titles and skills. This final model though rudimentary, allows us to give a basic notion of advice on skills that needs to be cultivated through comparison of a given skill-set and the skills within a particular job cluster.

Learning Points

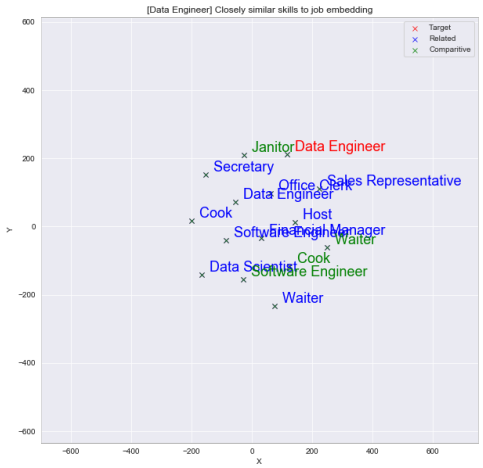
- Gathering real life data by building scraping tool
- Information retrieval technique such as TF-IDF
- Using Jaccard similarity for entity resolution in a non-trivial project
- Looking into and applying the nuances of Word2Vec and Doc2Vec and the various algorithms to train them
- Applying K-Means clustering in a non-trivial project along with analysis on unsupervised learning task
- Teamwork and communication

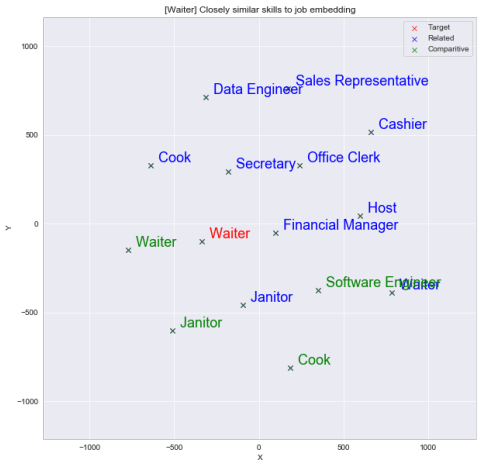
References

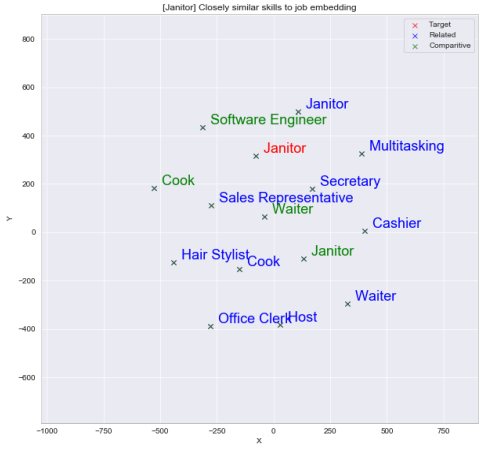
1. https://www.onetcenter.org/database.html
2. https://cs.stanford.edu/~quocle/paragraph_vector.pdf
3. https://en.wikipedia.org/wiki/Silhouette_(clustering)
4. https://medium.com/scaleabout/a-gentle-introduction-to-doc2vec-db3e8c0cce5e
5. https://stackoverflow.com/a/44013893/3289112
6. https://stats.stackexchange.com/a/299016

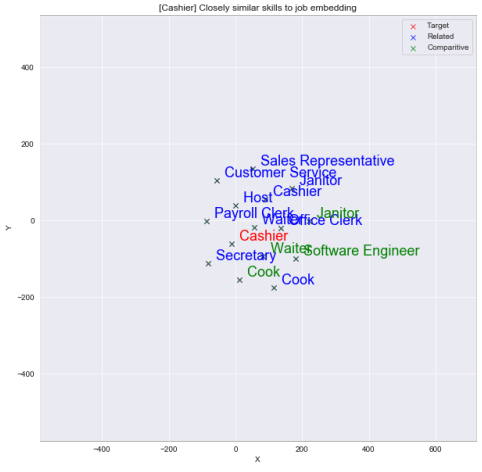
Appendix

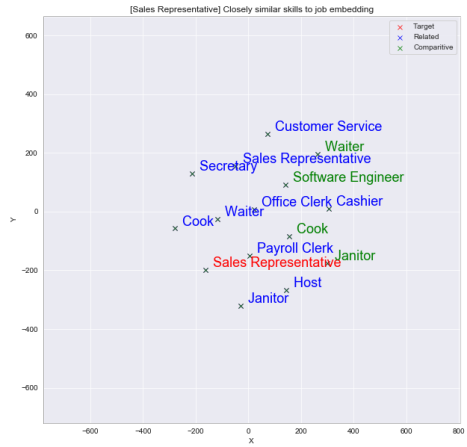
Appendix A: Top 10 most related skills to job title, calculated by cosine similarity of Word2Vec embeddings

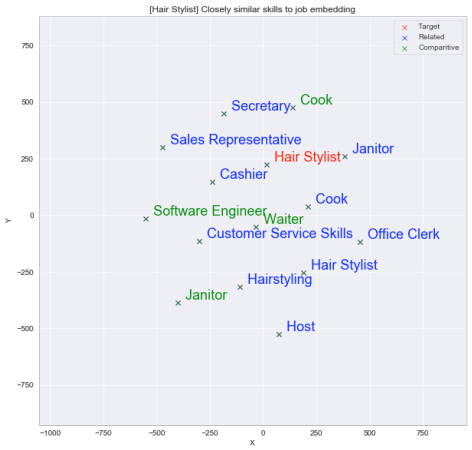




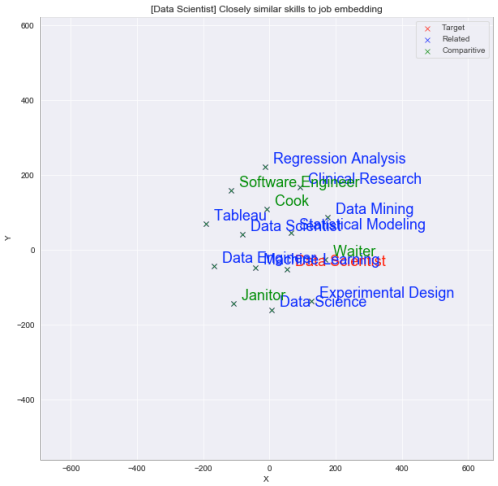


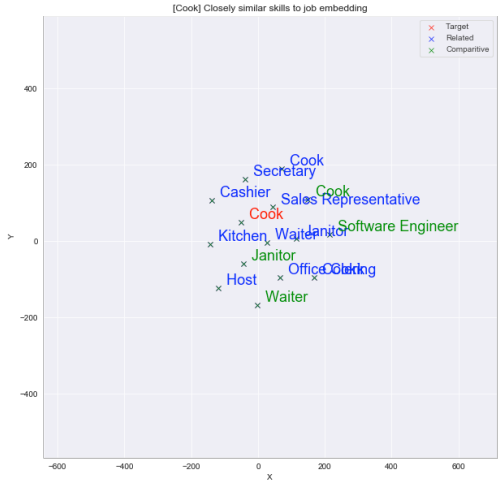



[Sales Representative] Closely similar skills to job embedding
Target
Related
Comparitive
Customer Service
Waiter
Sales Representative
Secretary
Software Engineer
Office Clerk
Cashier
Waiter
Cook
Cook
Payroll Clerk
Janitor
Sales Representative
Host
Janitor
X
Y

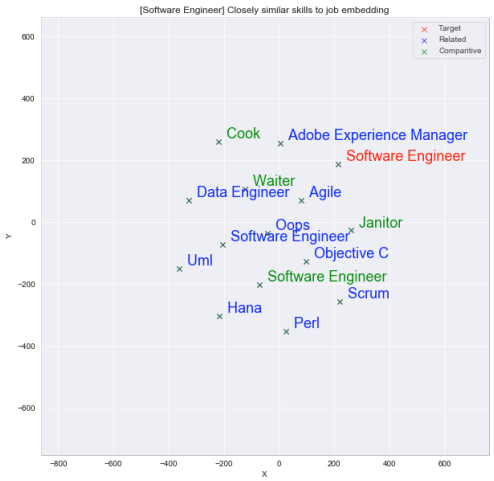

[Hair Stylist] Closely similar skills to job embedding
Target
Related
Comparitive
Secretary
Cook
Sales Representative
Janitor
Hair Stylist
Cashier
Cook
Software Engineer
Waiter
Customer Service Skills
Office Clerk
Hair Stylist
Hairstyling
Janitor
Host
X
Y

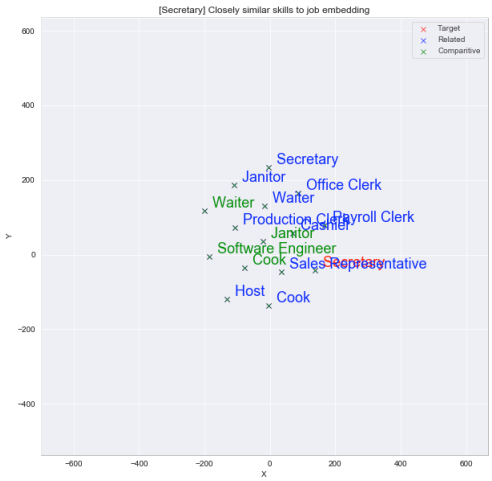

[Financial Manager] Closely similar skills to job embedding
Target
Related
Comparitive
Sales Representative
Cook
Waiter
Corporate Strategy
Financial Manager
Financial Manager
Portfolio Management
Office Clerk
Software Engineer
Janitor
Financial Planning
Payroll Clerk
X
Y

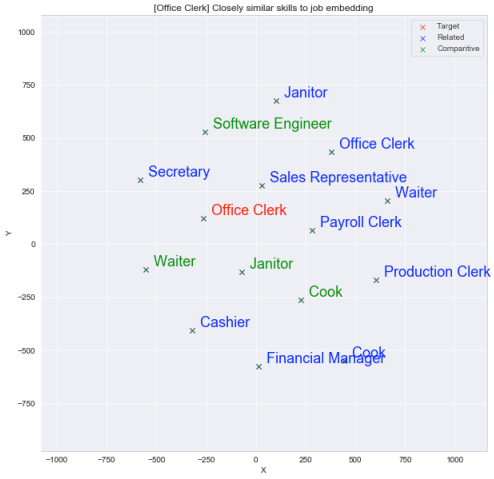

[Data Scientist] Closely similar skills to job embedding
Target
Related
Comparitive
Regression Analysis
Software Engineer
Clinical Research
Cook
Tableau
Data Mining
Statistical Modeling
Waiter
Janitor
Data Science
Experimental Design
X
Y

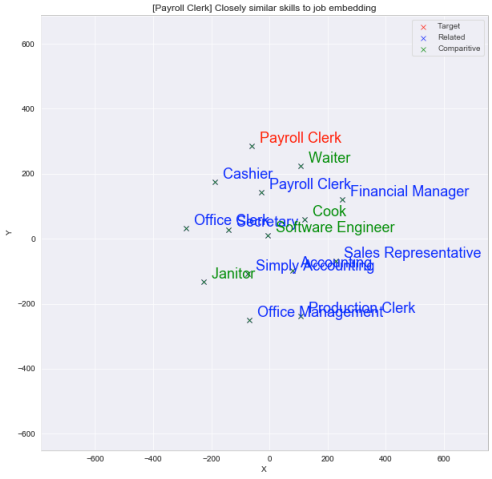

[Cook] Closely similar skills to job embedding
Target
Related
Comparitive
Cook
Secretary
Cashier
Cook
Sales Representative
Cook
Software Engineer
Kitchen
Waiter
Janitor
Janitor
Host
Waiter
X
Y

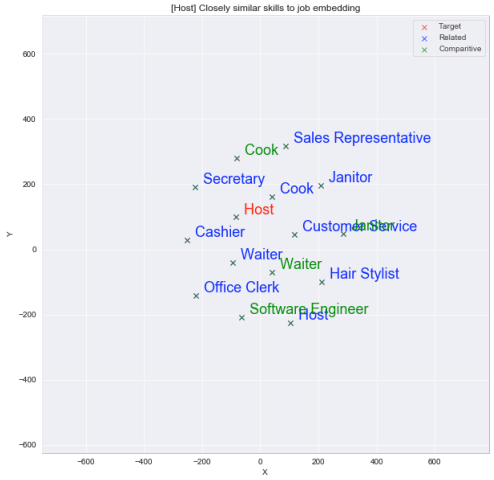

[Production Clerk] Closely similar skills to job embedding
Target
Related
Comparitive
Production Clerk
Keyboarding
Cashier
Secretary
Janitor
Cook
Payroll Clerk
Typing
Office Clerk
Janitor
Production Clerk
Sales Representative
Waiter
X
Y


[Software Engineer] Closely similar skills to job embedding
Target
Related
Comparitive
Cook
Adobe Experience Manager
Software Engineer
Waiter
Data Engineer
Agile
Oops
Janitor
Software Engineer
Objective C
Uml
Software Engineer
Scrum
Hana
Perl
X
Y

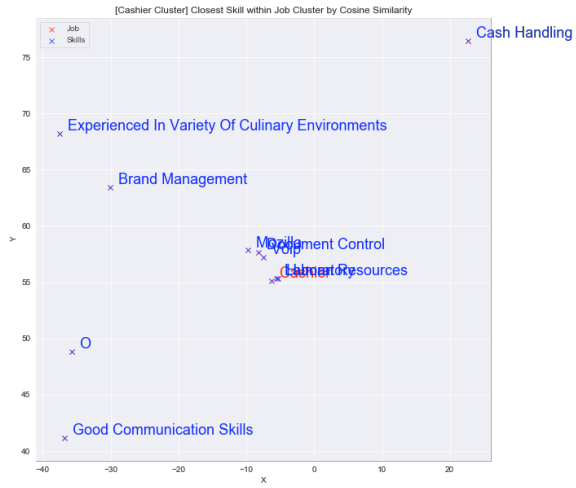

[Secretary] Closely similar skills to job embedding
Target
Related
Comparitive
Secretary
Janitor
Office Clerk
Waiter
Waiter
Payroll Clerk
Janitor
Software Engineer
Cook
Host
Cook
X
Y

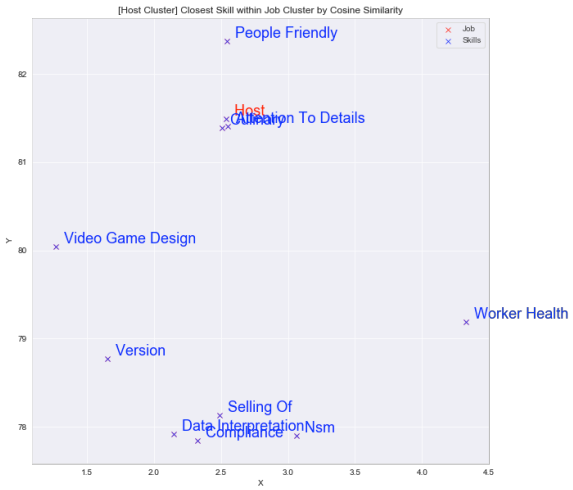

[Office Clerk] Closely similar skills to job embedding
Target
Related
Comparitive
Janitor
Software Engineer
Office Clerk
Secretary
Sales Representative
Waiter
Office Clerk
Payroll Clerk
Waiter
Janitor
Production Clerk
Cook
Cashier
Financial Manager
X
Y

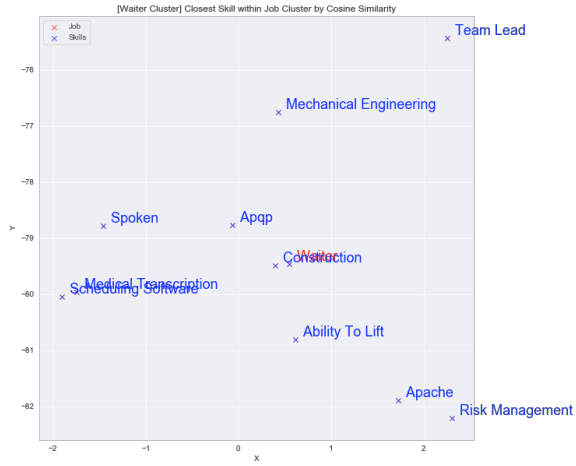

[Payroll Clerk] Closely similar skills to job embedding
Target
Related
Comparitive
Payroll Clerk
Waiter
Cashier
Payroll Clerk
Financial Manager
Cook
Software Engineer
Sales Representative
Janitor
Production Clerk
X
Y

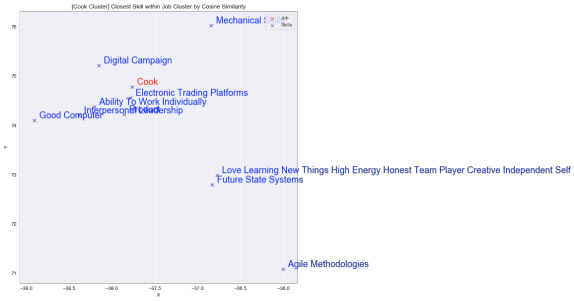

[Host] Closely similar skills to job embedding
Target
Related
Comparitive
Sales Representative
Cook
Secretary
Cook
Janitor
Host
Cashier
Waiter
Waiter
Hair Stylist
Office Clerk
Software Engineer
X
Y

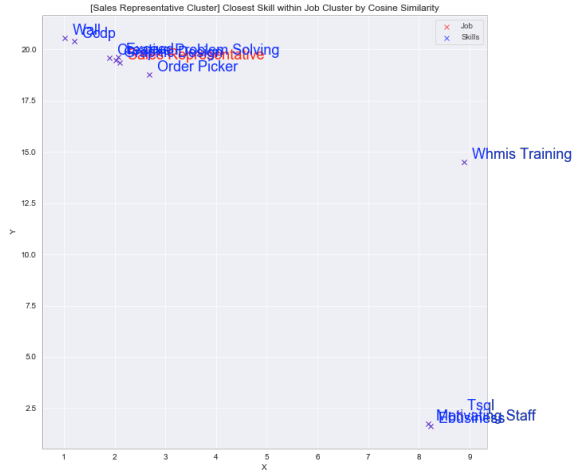
Appendix B: Clustered job and skills using Word2Vec embedding, showing at most top 10 most related skills for job title within job cluster by cosine

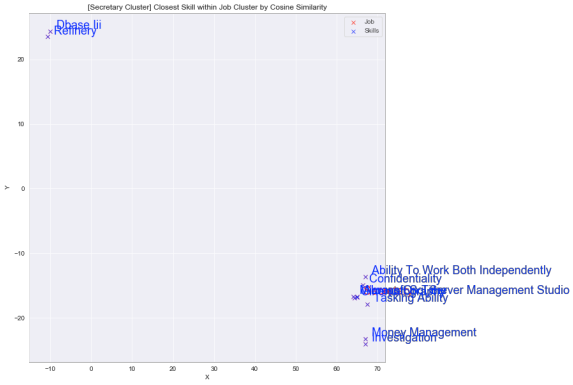


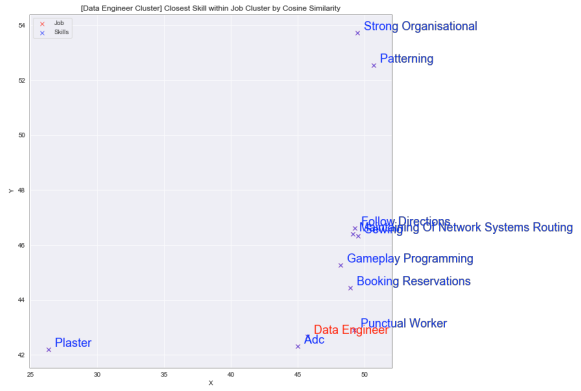


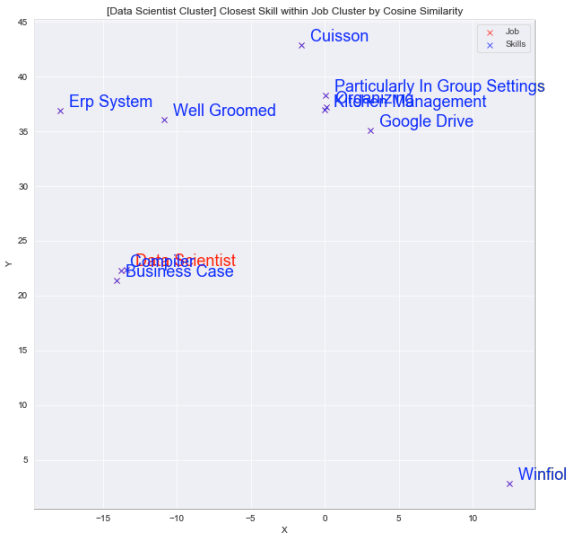


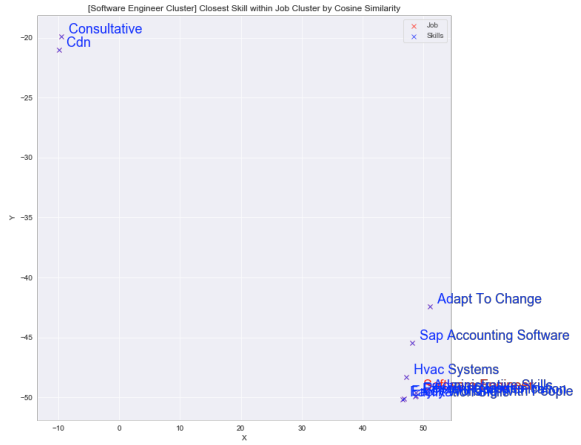


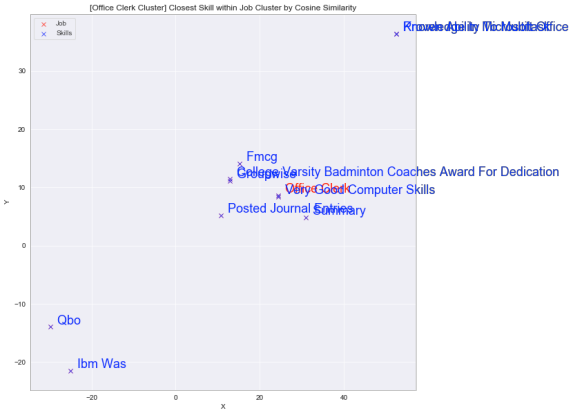




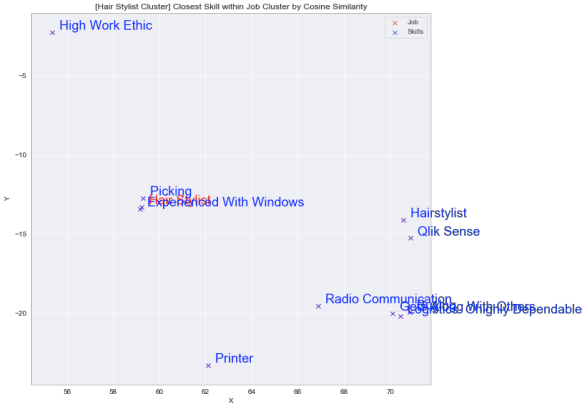

[Data Engineer Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Strong Organisational
Patterning
Follow Directions
Sewing
Gameplay Programming
Booking Reservations
Punctual Worker
Data Engineer
Adc
Plaster
X
Y

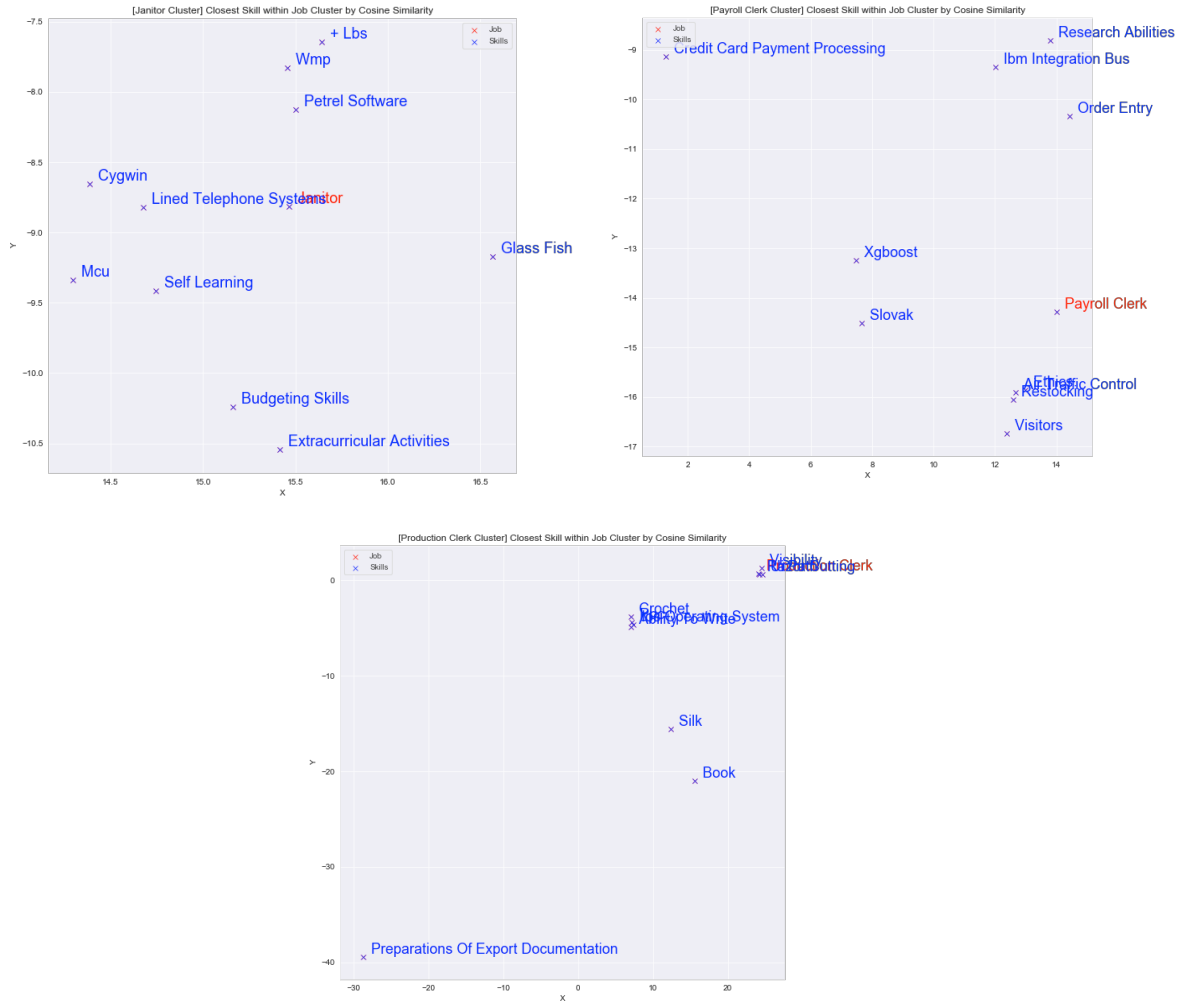

[Data Scientist Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Cuisson
Particularly In Group Settings
Erp System
Well Groomed
Google Drive
Business Case
Winfiol
X
Y


[Software Engineer Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Consultative
Cdn
Adapt To Change
Sap Accounting Software
Hvac Systems
X
Y

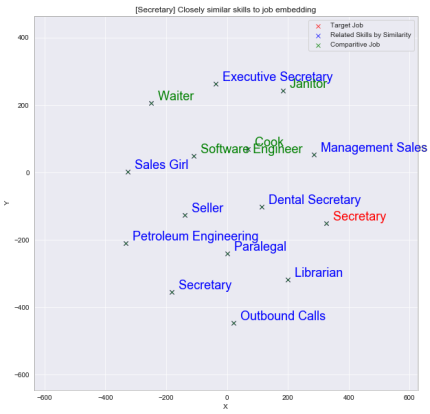

[Office Clerk Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Fmcg
Very Good Computer Skills
Posted Journal Entries
Qbo
Ibm Was
X
Y

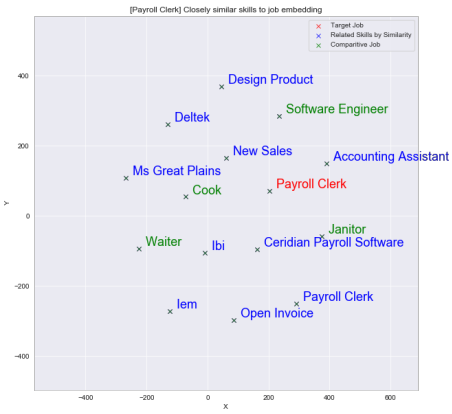

[Financial Manager Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Computer Typing
Clerical Duties
Skilled In Developing
In Person
Apply Document Control Processes
Published Author
Updos
X
Y


[Hair Stylist Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
High Work Ethic
Picking
Experienced With Windows
Hairstylist
Qlik Sense
Radio Communication
Printer
X
Y

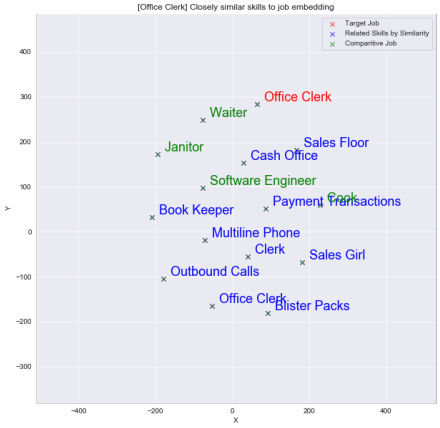


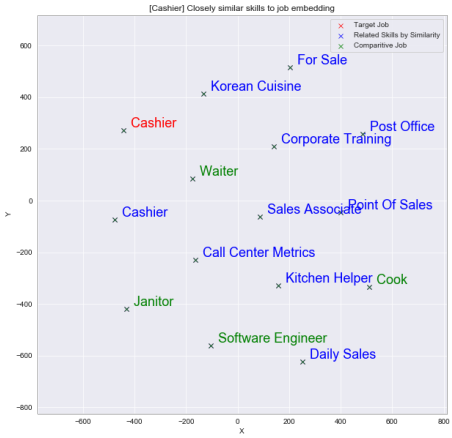
Appendix C: Top 10 most related similar skills to each job, calculated by cosine similarity of between job title and skill word embeddings

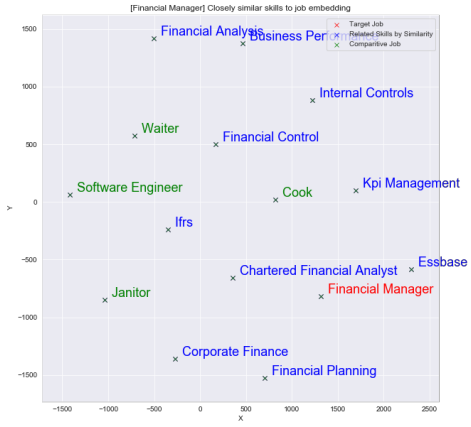

[Secretary] Closely similar skills to job embedding
Target Job
Related Skills by Similarity
Comparitive Job
Executive Secretary
Janitor
Waiter
Cook
Software Engineer
Management Sales
Sales Girl
Seller
Dental Secretary
Secretary
Petroleum Engineering
Paralegal
Librarian
Secretary
Outbound Calls
X
Y

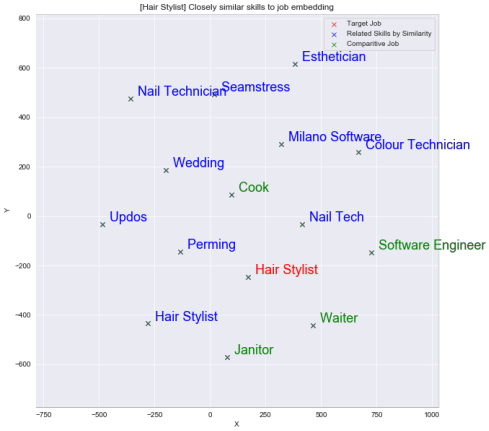

[Payroll Clerk] Closely similar skills to job embedding
Target Job
Related Skills by Similarity
Comparitive Job
Design Product
Deltek
Software Engineer
New Sales
Accounting Assistant
Ms Great Plains
Cook
Payroll Clerk
Janitor
Waiter
Ibi
Ceridian Payroll Software
Iem
Payroll Clerk
Open Invoice
X
Y

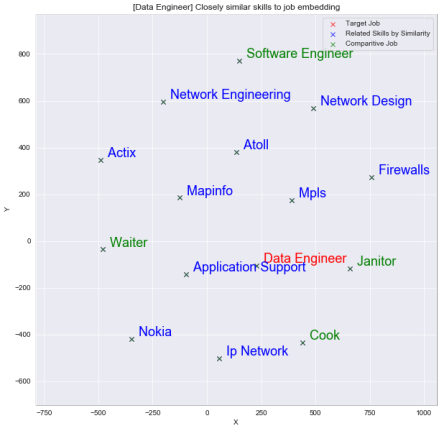

[Cook] Closely similar skills to job embedding
Target Job
Related Skills by Similarity
Comparitive Job
Cook
Cook
Housekeeper
Waiter
Meal Prep
Serving Customers
Bread
Cabinet Making
Cab Driver
Closing Duties
Software Engineer
Playing Musical Instruments
Kitchen Work
Cook
Janitor
X
Y

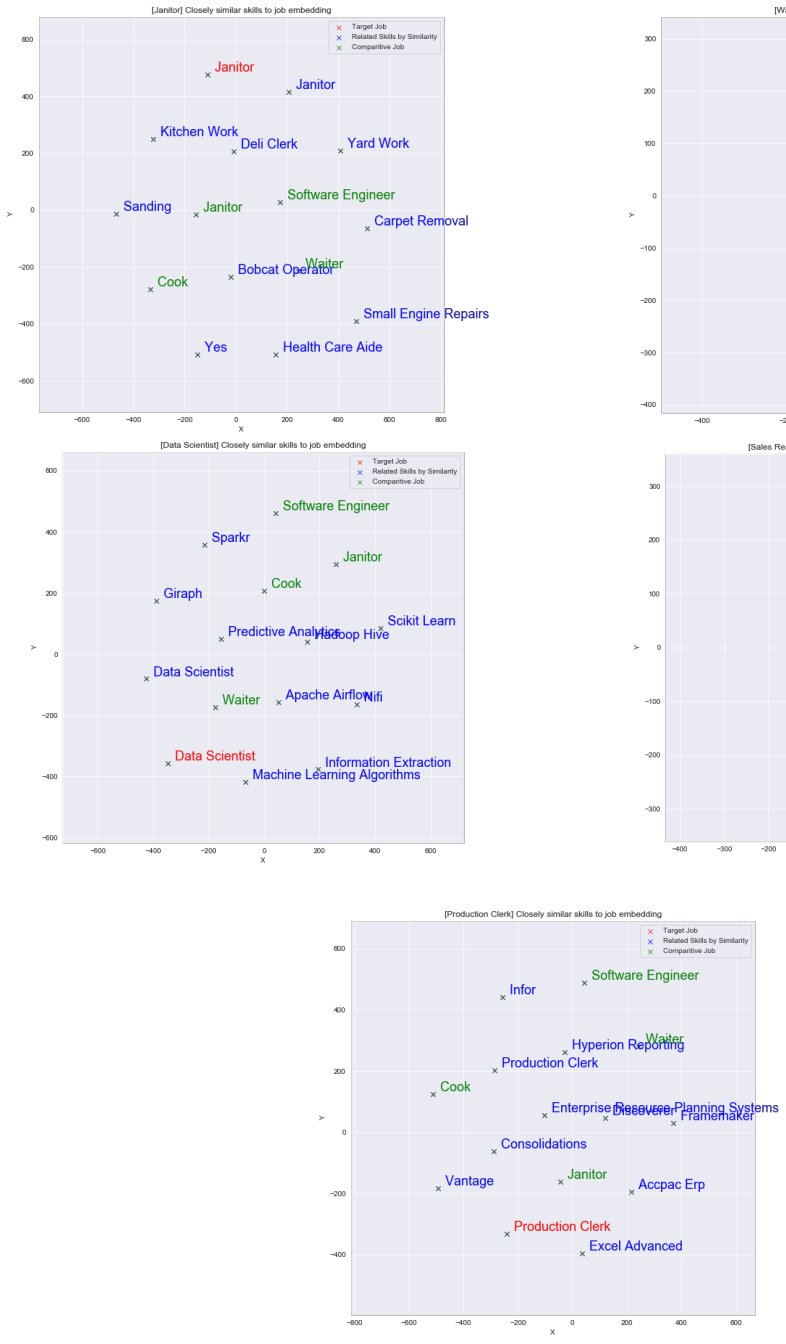

[Office Clerk] Closely similar skills to job embedding
Target Job
Related Skills by Similarity
Comparitive Job
Office Clerk
Waiter
Janitor
Sales Floor
Cash Office
Software Engineer
Cook
Payment Transactions
Book Keeper
Multiline Phone
Clerk
Sales Girl
Outbound Calls
Office Clerk
Blister Packs
X
Y


[Cashier] Closely similar skills to job embedding
Target Job
Related Skills by Similarity
Comparitive Job
For Sale
Korean Cuisine
Cashier
Post Office
Corporate Training
Waiter
Cashier
Sales Associate
Point Of Sales
Call Center Metrics
Kitchen Helper
Cook
Janitor
Software Engineer
Daily Sales
X
Y

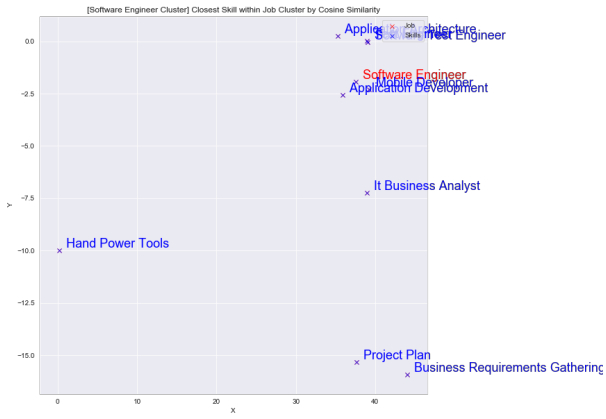

[Financial Manager] Closely similar skills to job embedding
Target Job
Related Skills by Similarity
Comparitive Job
Financial Analysis
Internal Controls
Waiter
Financial Control
Software Engineer
Cook
Kpi Management
Ifrs
Essbase
Chartered Financial Analyst
Janitor
Financial Manager
Corporate Finance
Financial Planning
X
Y

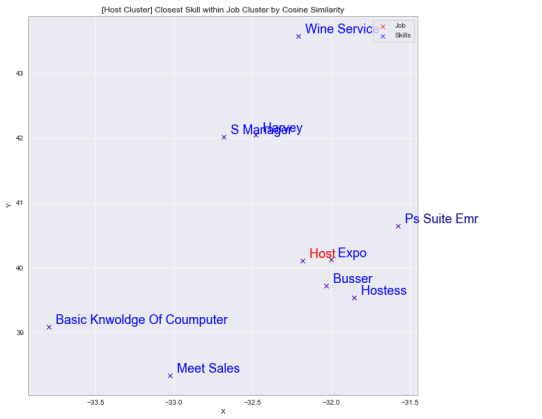

[Hair Stylist] Closely similar skills to job embedding
Target Job
Related Skills by Similarity
Comparitive Job
Esthetician
Nail Technician
Seamstress
Milano Software
Colour Technician
Wedding
Cook
Updos
Nail Tech
Perming
Software Engineer
Hair Stylist
Hair Stylist
Waiter
Janitor
X
Y

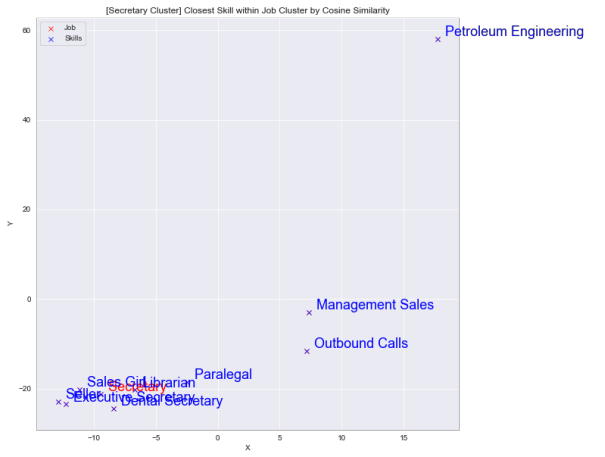

[Data Engineer] Closely similar skills to job embedding
Target Job
Related Skills by Similarity
Comparitive Job
Software Engineer
Network Engineering
Network Design
Actix
Atoll
Firewalls
Mapinfo
Mpls
Waiter
Application Support
Data Engineer
Janitor
Nokia
Cook
Ip Network
X
Y

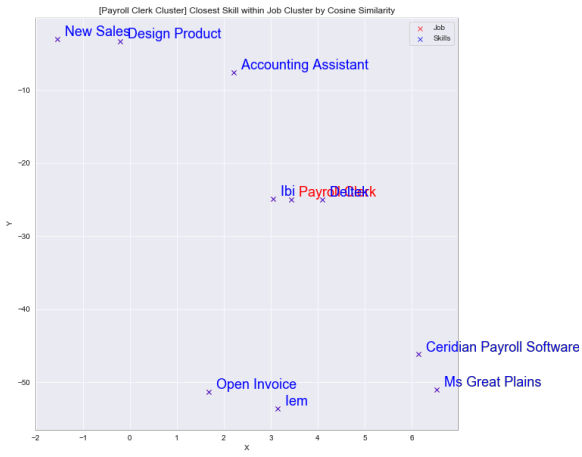


Appendix D: Clustered job and skills using Doc2Vec embedding for job titles and Word2Vec embedding for skills, showing at most top 10 most related skills for job title within job cluster by cosine similarity

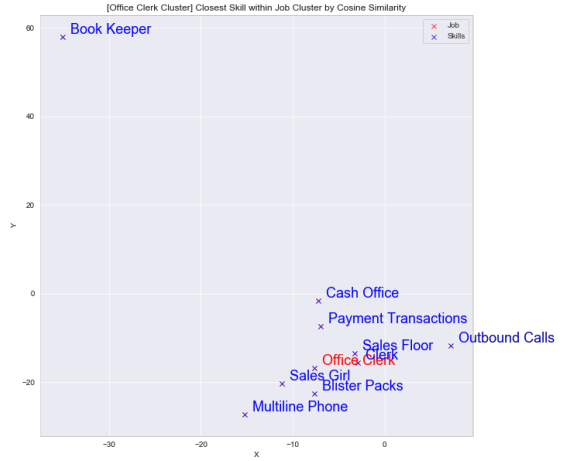

[Software Engineer Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Software Engineer
Mobile Developer
Application Development
It Business Analyst
Hand Power Tools
Project Plan
Business Requirements Gathering
X
Y

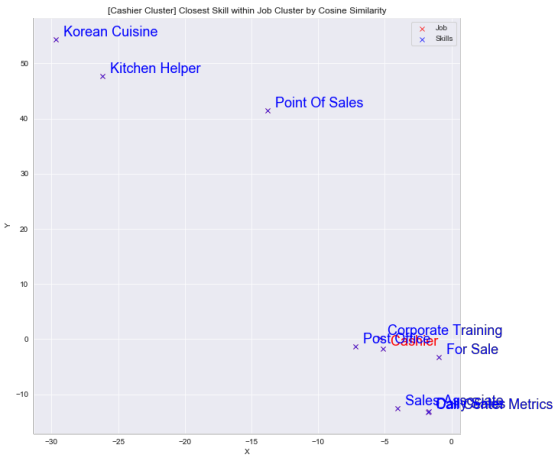

[Host Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Wine Servic
S Manager
Ps Suite Emr
Host Expo
Busser
Hostess
Basic Knwoldge Of Coumputer
Meet Sales
X
Y


[Secretary Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Petroleum Engineering
Management Sales
Outbound Calls
Paralegal
Sales Girl
Librarian
Executive Secretary
Dental Secretary
X
Y

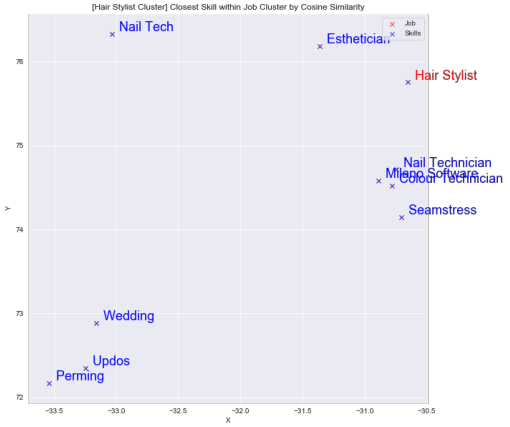

[Payroll Clerk Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
New Sales
Design Product
Accounting Assistant
Ceridian Payroll Software
Open Invoice
Ms Great Plains
X
Y

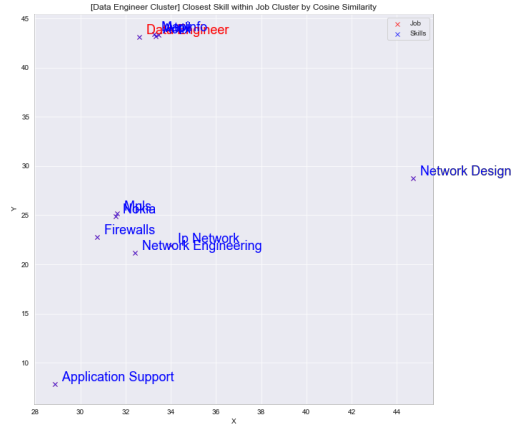

[Cook Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Bread
Cabinet Making
Kitchen Work
Closing Duties
X
Y

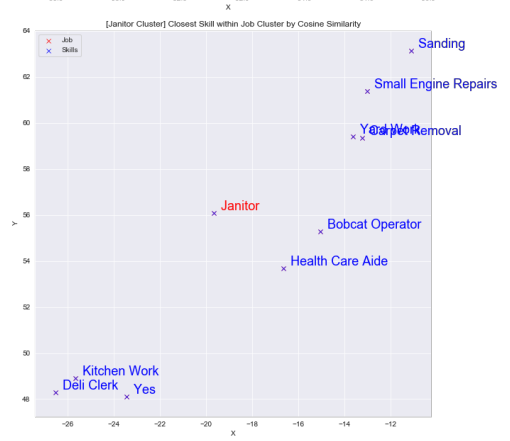

[Office Clerk Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Book Keeper
Cash Office
Payment Transactions
Outbound Calls
Sales Floor
Clerk
Office Clerk
Sales Girl
Blister Packs
Multiline Phone
X
Y

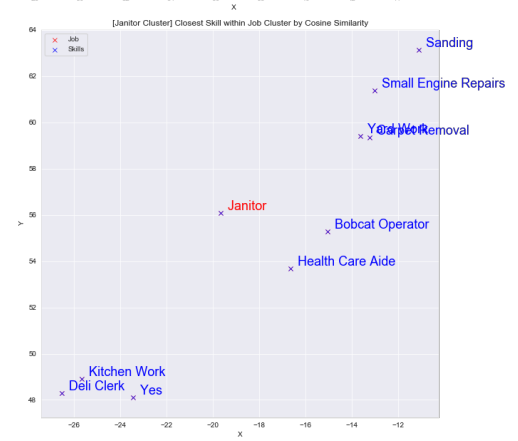

[Cashier Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Korean Cuisine
Kitchen Helper
Point Of Sales
Corporate Training
Cashier
For Sale
Metrics
X
Y

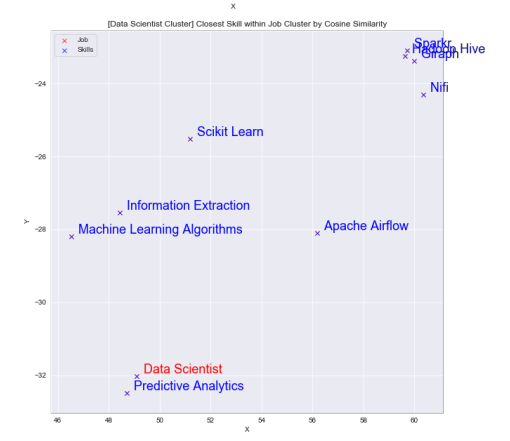

[Financial Manager Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Financial Planning
Kpi Management
Financial Control
Financial Analysis
Business Performance
Essbase
X
Y


[Hair Stylist Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Nail Tech
Hair Stylist
Nail Technician
Seamstress
Wedding
Updos
Perming
X
Y

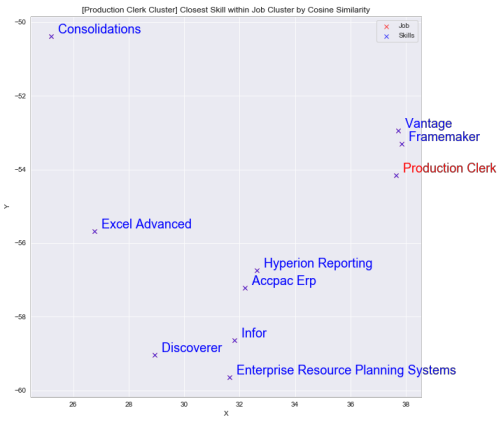

[Data Engineer Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Network Design
Firewalls
Network Engineering
Application Support
X
Y


[Janitor Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Sanding
Small Engine Repairs
Janitor
Bobcat Operator
Health Care Aide
Kitchen Work
Deli Clerk
Yes
X
Y


[Janitor Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Sanding
Small Engine Repairs
Janitor
Bobcat Operator
Health Care Aide
Kitchen Work
Deli Clerk
Yes
X
Y


[Data Scientist Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Nifi
Scikit Learn
Information Extraction
Machine Learning Algorithms
Apache Airflow
Data Scientist
Predictive Analytics
X
Y


[Sales Representative Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
As Sales
Management Sales
Sales And
Sales In
Of Sales
And Selling
X
Y


[Production Clerk Cluster] Closest Skill within Job Cluster by Cosine Similarity
Job
Skills
Consolidations
Vantage
Framemaker
Production Clerk
Excel Advanced
Hyperion Reporting
Accpac Erp
Infor
Discoverer
Enterprise Resource Planning Systems
−50
−52
−54
−56
−58
−60
Y
26
28
30
32
34
36
38
X