Building a Skills Taxonomy

Building a Skills Taxonomy.
India Kerle

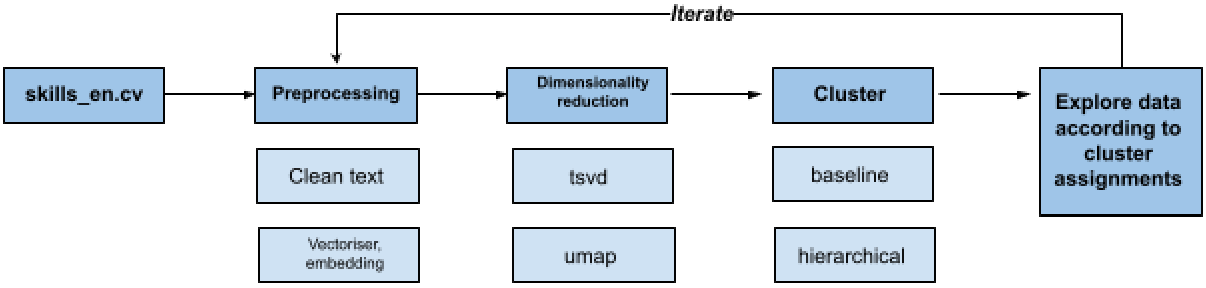
How I made sense of and approached the problem

I approached this problem following the CRISP-DM framework for structuring data analysis tasks. Following this, my first step was to develop an understanding of the problem, the data and possible data-driven methods for approaching the problem. I did so by:

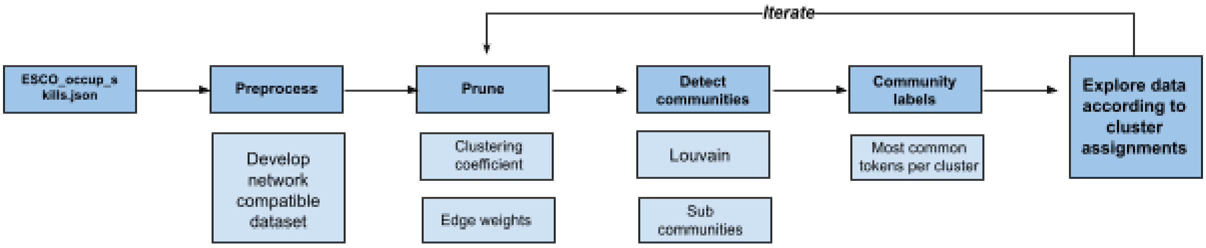
1. **Reading around:** I read papers on building alternative skills taxonomies to get a better sense of the problem and of how others have already approached the issue.
2. **Investigating the data:** I made notes of what every column in **skills_en.csv** and the other data provided was; making use of the ESCO Service Platform model codebook to clarify terms.

Based on my initial investigation, I developed **two** different process diagrams that could theoretically answer the problem using NLP or network analysis.

1. A purely NLP approach



2. A purely network analysis approach



I attempted both process possibilities to address the problem and to build a more in depth understanding of the data.

How I built an understanding of the data

In addition to understanding the data substantively, I ran basic **exploratory data analysis** of the skills_en.csv dataset to get a sense of **data completeness, patterns in the text** and **general distributions** of the data:

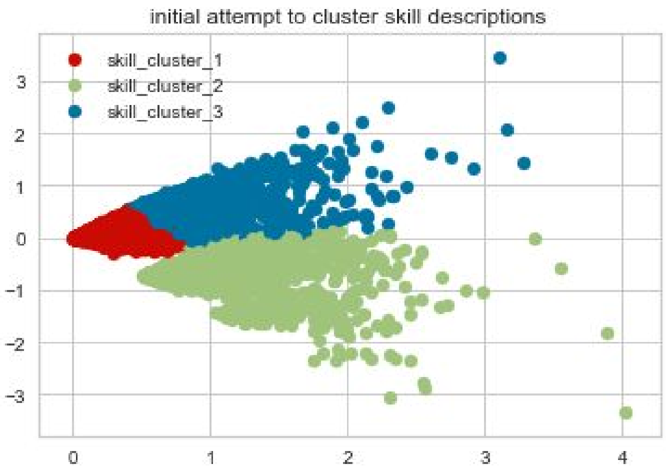
1. **Data completeness:** I explored if the data had NAs, if any text columns needed cleaning, if any skills were labelled i.e. *reusabilityLevel*, *skillsLevel* with wrong categories and verified that values in the *preferredLabel* column were indeed unique

2. **Text patterns:** I investigated the average length of skills descriptions, types of terms used (verbs, nouns etc.) per text column using POS tagging, average length of total text and types per text

3. **General distributions:** I explored how many skills belonged to different reusability levels and skill types by generating simple bar charts

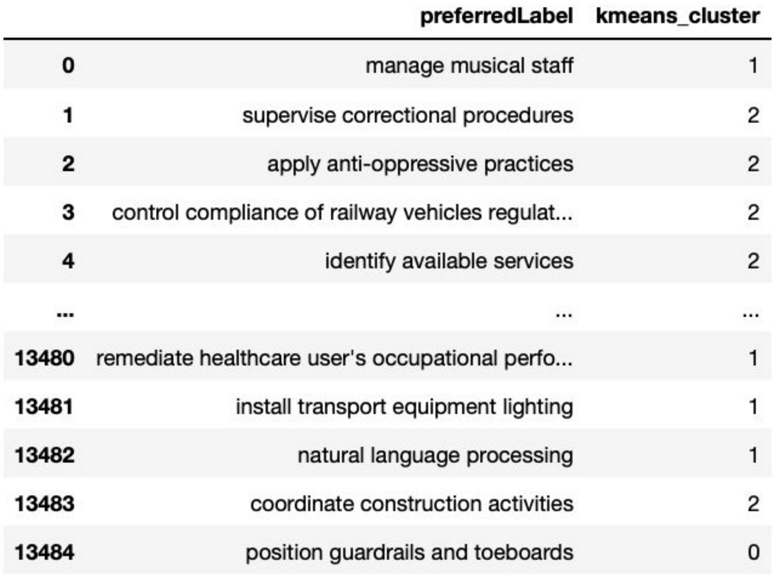
I also developed a better sense of the data by approaching the problem as a purely NLP problem.

As a **baseline model**, I concatenated all text columns, preprocessed the total text to: a) remove stopwords, b) remove numbers, c) remove punctuation and lemmatise the terms, (d) converted the text to a count vectoriser, (e) reduced its dimensionality via Truncated Singular Value Decomposition (tsvd) and ran kmeans (k = 3) to explore what the data looked like in a vector space and how well a simple clustering algorithm would perform.



K-Means clusters on a tsvd-reduced count vectoriser of skills text

I then assigned a skill to each cluster to investigate whether the clusters made sense by spot checking results.



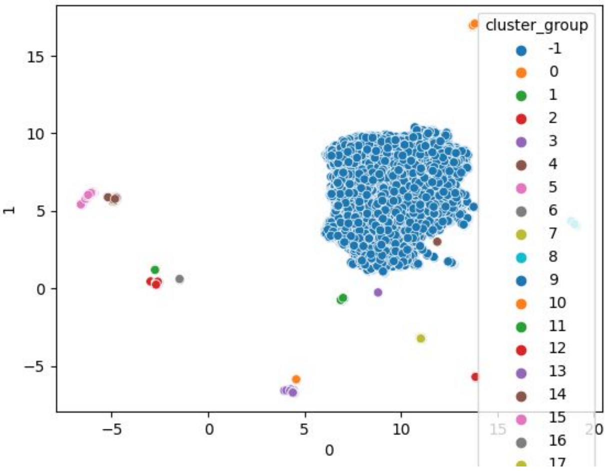
	preferredLabel	kmeans_cluster
0	manage musical staff	1
1	supervise correctional procedures	2
2	apply anti-oppressive practices	2
3	control compliance of railway vehicles regulat...	2
4	identify available services	2
...	...	...
13480	remediate healthcare user's occupational perfo...	1
13481	install transport equipment lighting	1
13482	natural language processing	1
13483	coordinate construction activities	2
13484	position guardrails and toeboards	0

Skill names and their associated K-Means cluster

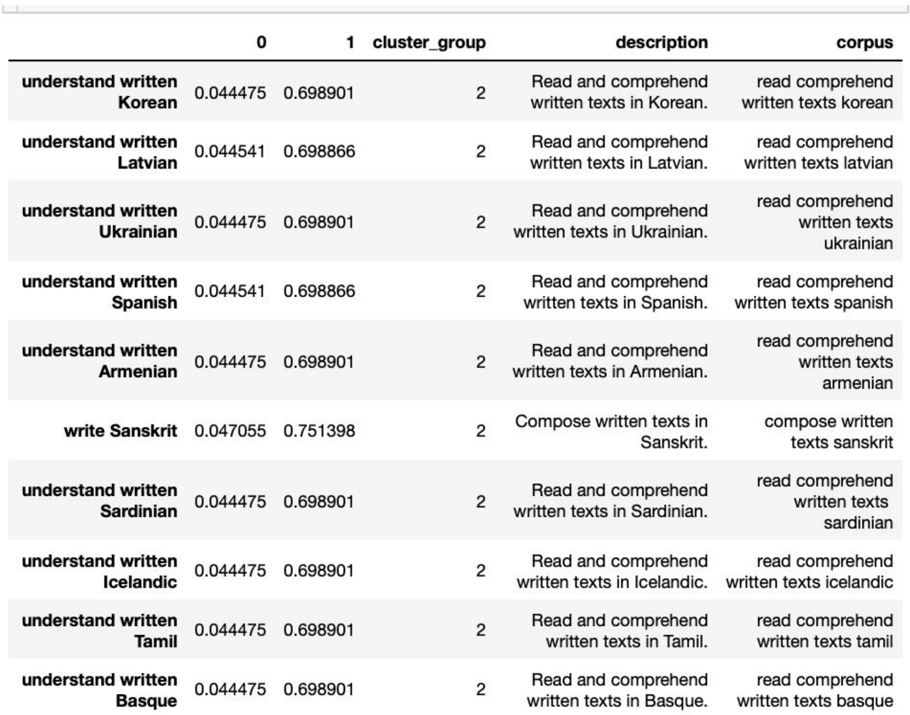
It appeared immediately clear that the clusters were mostly nonsensical so I tinkered with different ways of:

1. **Preprocessing the text:** only including POS tagged nouns as the text data, getting rid of the top 50 most common tokens, removing almost repeat sentences, using a tfidf-vectoriser, using word embeddings (Word2Vec, Spacy, fasttext)
2. **Reducing dimensionality:** using Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) instead of TSVD
3. **Using clustering algorithms:** hierarchical clustering, DBSCAN

Whilst these experiments improved upon the baseline model and almost all combinations of attempts were successful at pulling highly specialised, almost identically-described skills, they did not do a good job of pulling apart the vast majority of unclustered skill vectors (or embeddings, see below as the large unclustered -1 cluster group):



DBSCAN clusters on tsvd-reduced tfidf skills vectors



	0	1	cluster_group	description	corpus
understand written Korean	0.044475	0.698901	2	Read and comprehend written texts in Korean.	read comprehend written texts korean
understand written Latvian	0.044541	0.698866	2	Read and comprehend written texts in Latvian.	read comprehend written texts latvian
understand written Ukrainian	0.044475	0.698901	2	Read and comprehend written texts in Ukrainian.	read comprehend written texts ukrainian
understand written Spanish	0.044541	0.698866	2	Read and comprehend written texts in Spanish.	read comprehend written texts spanish
understand written Armenian	0.044475	0.698901	2	Read and comprehend written texts in Armenian.	read comprehend written texts armenian
write Sanskrit	0.047055	0.751398	2	Compose written texts in Sanskrit.	compose written texts sanskrit
understand written Sardinian	0.044475	0.698901	2	Read and comprehend written texts in Sardinian.	read comprehend written texts sardinian
understand written Icelandic	0.044475	0.698901	2	Read and comprehend written texts in Icelandic.	read comprehend written texts icelandic
understand written Tamil	0.044475	0.698901	2	Read and comprehend written texts in Tamil.	read comprehend written texts tamil
understand written Basque	0.044475	0.698901	2	Read and comprehend written texts in Basque.	read comprehend written texts basque

Clustering did a good job of labelling similarly written skills descriptions but otherwise fell short to pull apart the majority of less-specific descriptions

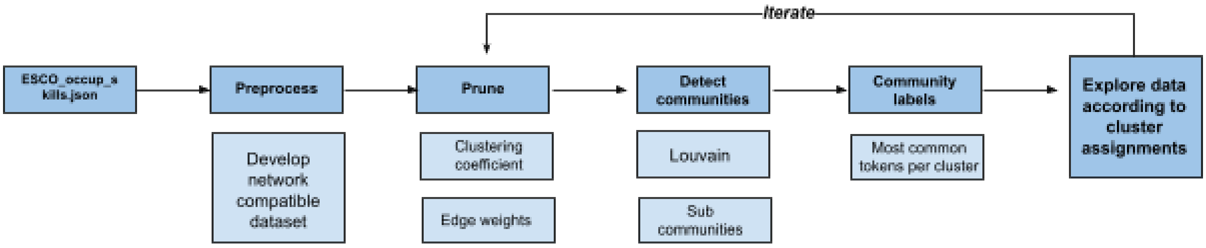
Via a purely NLP approach, I learned that the skills text data:

- Had descriptions that were identically phrased
- Had many descriptions that were very vague while others were very specific
- Clustering algorithms did a good job of clustering skills based on a single term

While it may well be possible to further disaggregate the central 'blob' by improving the NLP pipeline (e.g. possibly a more sophisticated word embedding could pull out different semantically related groups), these initial conclusions were enough to suggest taking a different approach was worth exploring as creating a robust taxonomy with such vague descriptions seemed unlikely.

Based on this better understanding of the data however, I therefore decided to approach the problem using the second process framework, a purely network analysis approach.

A thought-through technical approach to solving the problem



The final technical approach

To approach this problem using network analysis techniques I deployed the following steps:

Preprocess

I first reformatted the *ESCO_occup_skills.json* data ready for network analysis. I did so by generating all pair combinations of skills within the job occupation *hasEssentialSkill* dictionary values. I then counted the number of skills that co-occurred in an occupation. I ultimately generated a dataframe where column 1 represented skill 1, column 2 represented skill 2, and column 3 represented the number of jobs that the two skills co-occurred in. Each skill therefore represented nodes in the network and the number of jobs that each skill shared represented the weight of the edges between nodes in the network.

Prune

Without any pruning, the network contained 10,536 nodes (skills) and 355,973 edges (weighted by number of jobs), including skills of every reusability level. According to Djumalieva and Sleeman (2018), 'the presence of highly central skills engenders one giant community that contains most skills and very few specialised communities.'' Skills that are very generic i.e. "are relevant to a broad range of occupations and sectors" (ESCO, n.d.) would otherwise obscure communities that were substantively meaningful.

I therefore pruned the network via two ways:

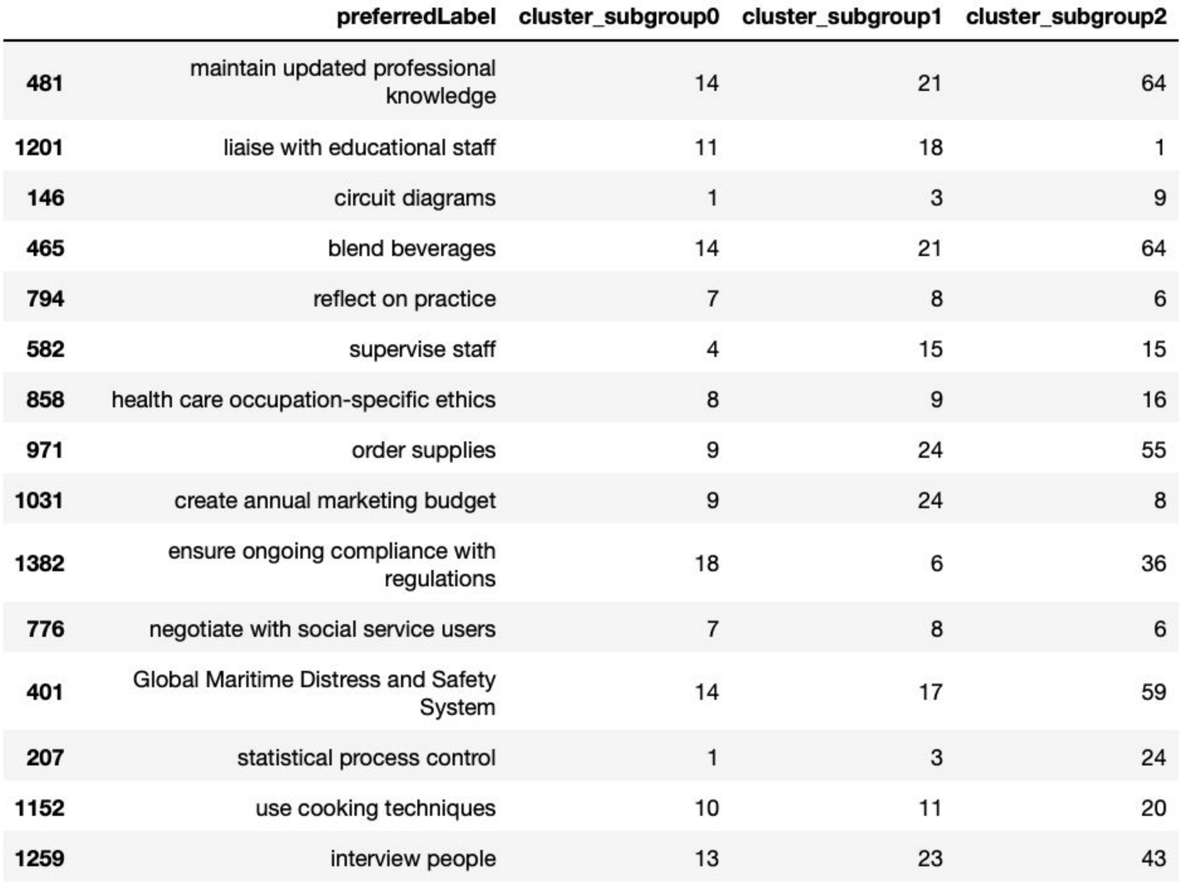
1. **Removing nodes based on edge weight:** I subsetted the dataframe to only include nodes with weight edges of more than 5 to remove skills that co-occur often with any other skills as these are unlikely to be useful to find communities.
2. **Clustering coefficients:** in line with Djumalieva and Sleeman (2018), I calculated the clustering coefficient of each node and removed nodes with clustering coefficients of less than 0.1 (based on looking at the coefficient histogram).

This resulted in a network of theoretically more specialised skills, with 1,357 nodes and 13,385 edges.

Detect Communities

In order to detect **meaningful groups of skills,** I ran a Louvain community detection algorithm. The algorithm works by splitting a network into mutually exclusive communities such that the number of edges *between* different communities is less than expected and the number of edges *within* a community is higher than expected compared to a randomly connected graph. The algorithm's default resolution parameter (resolution = 1) is already optimised to maximize modularity, a measurement from -1 to 1 of how well a community structure is formed (Gandhi, 2019).

I further deployed an extension to this algorithm to produce hierarchical communities using a dendrogram tree approach to partitioning. Using this technique I was able to detect 23 first level communities, 26 second level communities and 68 third level communities:



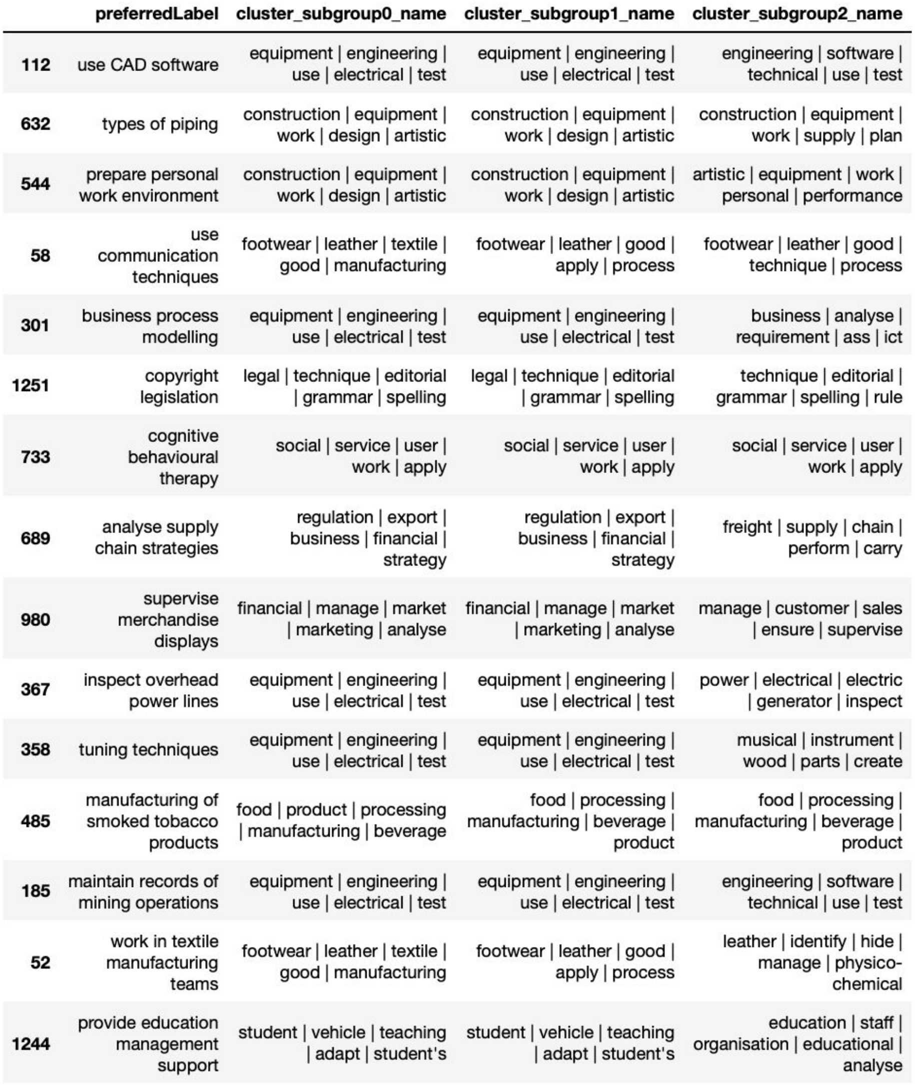
	preferredLabel	cluster_subgroup0	cluster_subgroup1	cluster_subgroup2
481	maintain updated professional knowledge	14	21	64
1201	liaise with educational staff	11	18	1
146	circuit diagrams	1	3	9
465	blend beverages	14	21	64
794	reflect on practice	7	8	6
582	supervise staff	4	15	15
858	health care occupation-specific ethics	8	9	16
971	order supplies	9	24	55
1031	create annual marketing budget	9	24	8
1382	ensure ongoing compliance with regulations	18	6	36
776	negotiate with social service users	7	8	6
401	Global Maritime Distress and Safety System	14	17	59
207	statistical process control	1	3	24
1152	use cooking techniques	10	11	20
1259	interview people	13	23	43

A sample of skill names and their associated subgroups

Community Labels

Finally, in order to label the communities, I first created a dataframe of all unique skill names (*preferredLabel*) and their associated community and subcommunities. I then preprocessed the *preferredLabel* text and found the top five cleaned tokens per community. I used top key terms as a label for every community.

Although going forward a more sophisticated technique to derive cluster names may be helpful (e.g. topic modelling on the full skill descriptions or a combined technique using qualitative approaches as well as data-driven approaches), for this initial analysis the top token approach seemed to give reasonable, understandable results.



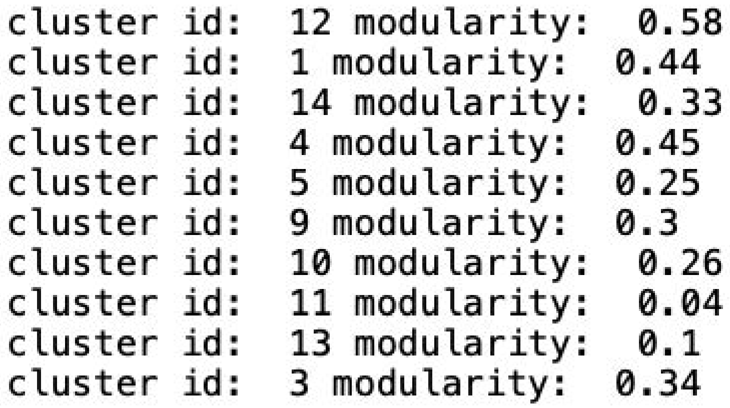
	preferredLabel	cluster_subgroup0_name	cluster_subgroup1_name	cluster_subgroup2_name
112	use CAD software	equipment | engineering | use | electrical | test	equipment | engineering | use | electrical | test	engineering | software | technical | use | test
632	types of piping	construction | equipment | work | design | artistic	construction | equipment | work | design | artistic	construction | equipment | work | supply | plan
544	prepare personal work environment	construction | equipment | work | design | artistic	construction | equipment | work | design | artistic	artistic | equipment | work | personal | performance
58	use communication techniques	footwear | leather | textile | good | manufacturing	footwear | leather | good | apply | process	footwear | leather | good | technique | process
301	business process modelling	equipment | engineering | use | electrical | test	equipment | engineering | use | electrical | test	business | analyse | requirement | ass | ict
1251	copyright legislation	legal | technique | editorial | grammar | spelling	legal | technique | editorial | grammar | spelling	technique | editorial | grammar | spelling | rule
733	cognitive behavioural therapy	social | service | user | work | apply	social | service | user | work | apply	social | service | user | work | apply
689	analyse supply chain strategies	regulation | export | business | financial | strategy	regulation | export | business | financial | strategy	freight | supply | chain | perform | carry
980	supervise merchandise displays	financial | manage | market | marketing | analyse	financial | manage | market | marketing | analyse	manage | customer | sales | ensure | supervise
367	inspect overhead power lines	equipment | engineering | use | electrical | test	equipment | engineering | use | electrical | test	power | electrical | electric | generator | inspect
358	tuning techniques	equipment | engineering | use | electrical | test	equipment | engineering | use | electrical | test	musical | instrument | wood | parts | create
485	manufacturing of smoked tobacco products	food | product | processing | manufacturing | beverage	food | processing | manufacturing | beverage | product	food | processing | manufacturing | beverage | product
185	maintain records of mining operations	equipment | engineering | use | electrical | test	equipment | engineering | use | electrical | test	engineering | software | technical | use | test
52	work in textile manufacturing teams	footwear | leather | textile | good | manufacturing	footwear | leather | good | apply | process	leather | identify | hide | manage | physico-chemical
1244	provide education management support	student | vehicle | teaching | adapt | student's	student | vehicle | teaching | adapt | student's	education | staff | organisation | educational | analyse

A sample of skill names and their associated subgroup names

Your ability to assess your solution

I assessed the 'meaningfulness' of skills communities along two evaluative measures:

1. **Modularity Scores:** Modularity refers to how well a community structure is formed. The higher the score is, the 'better' the result should theoretically be. I calculated the network's overall modularity score (**0.68**) and each top community's modularity score:



```
cluster id:  12 modularity:  0.58
cluster id:  1 modularity:  0.44
cluster id:  14 modularity:  0.33
cluster id:  4 modularity:  0.45
cluster id:  5 modularity:  0.25
cluster id:  9 modularity:  0.3
cluster id:  10 modularity:  0.26
cluster id:  11 modularity:  0.04
cluster id:  13 modularity:  0.1
cluster id:  3 modularity:  0.34
```

I investigated communities with particularly high and particularly low modularity scores (less than 0.3) and found that these confirmed the intuition that the lower modularity communities had less well defined sub clusters, while the high modularity communities had very distinct clusters. Further investigation into how to improve the modularity score of these less well defined communities would be a promising line of investigation going forward.

2. **Spot checking:** I also manually read skill names and descriptions per community and sub community to qualitatively assess if they made sense together. While a few communities with particularly low modularity scores had some skills that didn't seem to fit that well together, the vast majority of communities appeared meaningful and included groups of skills that 'made sense' to be clustered together.

What I would do next time

While the above technical approach produced a meaningful taxonomy of skills, going forward i would address the following:

1. **Not all skills were included:** I only included *'hasEssentialskill'* in the network and I pruned quite a few skills based on edge weight. Going forward, I would explore the impact on also using optional skills to build the network, as well as different approaches to pruning skills based on edge weight etc. to see if more skills could be included while maintaining community modularity scores.

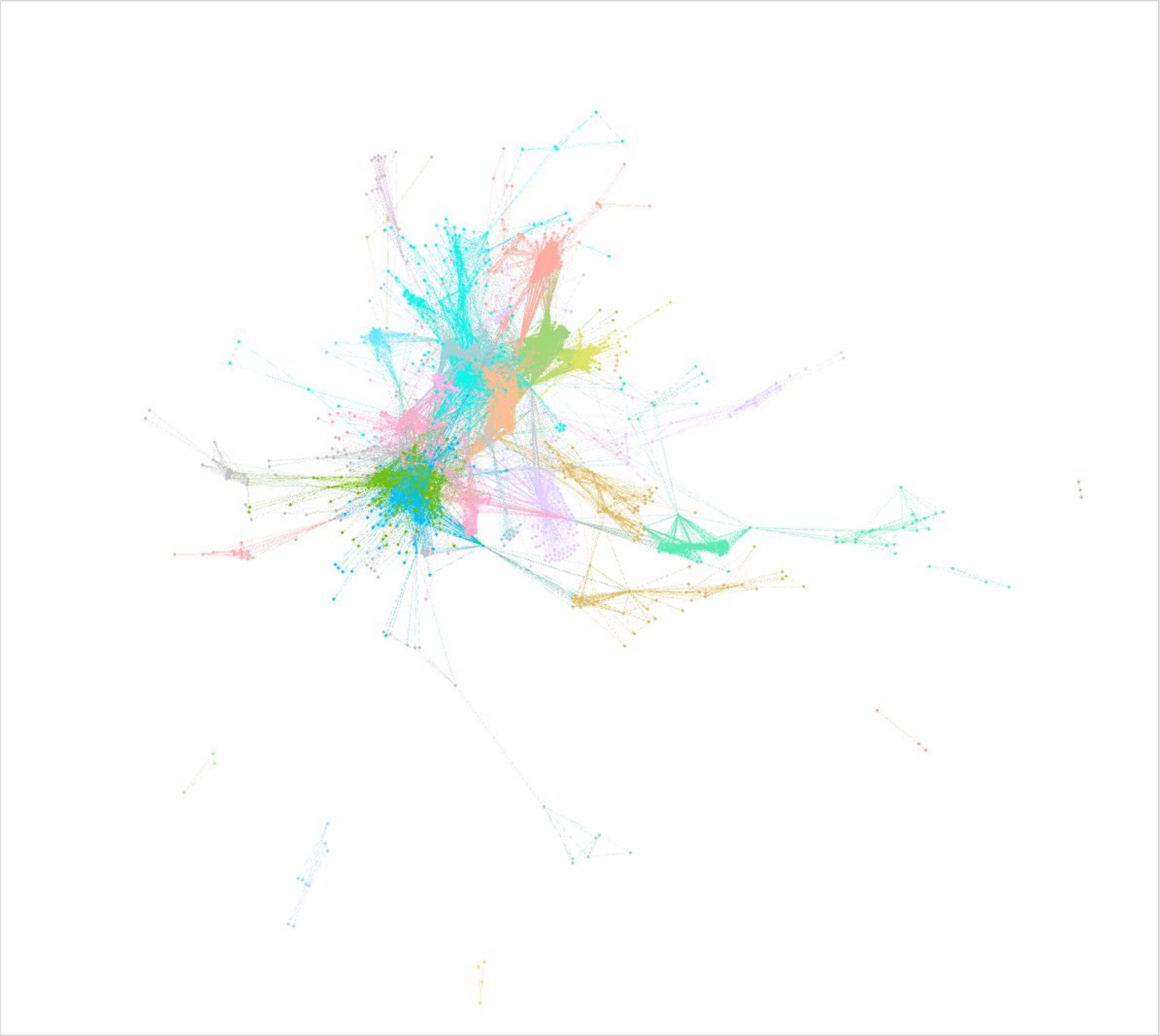
2. **Improve communities with low modularity scores:** Looking further into additional node/edge attributes that could help increase signal in the network as well as deriving more sophisticated network analysis metrics could help improve communities modularity score and improve the taxonomies robustness.

3. **Explore Network Analysis and NLP hybrid approaches:** I would also explore the role NLP methodologies could have in improving the network. In particular, I would be interested in either using NLP to add more sophisticated edge attributes to the network analysis (e.g. cosine similarity of skill pairs from an word embedding space to adjust the edge weights) or using NLP to find connections between smaller clusters which are not linked by edges and thus are difficult to fit into the taxonomy using network analysis alone.

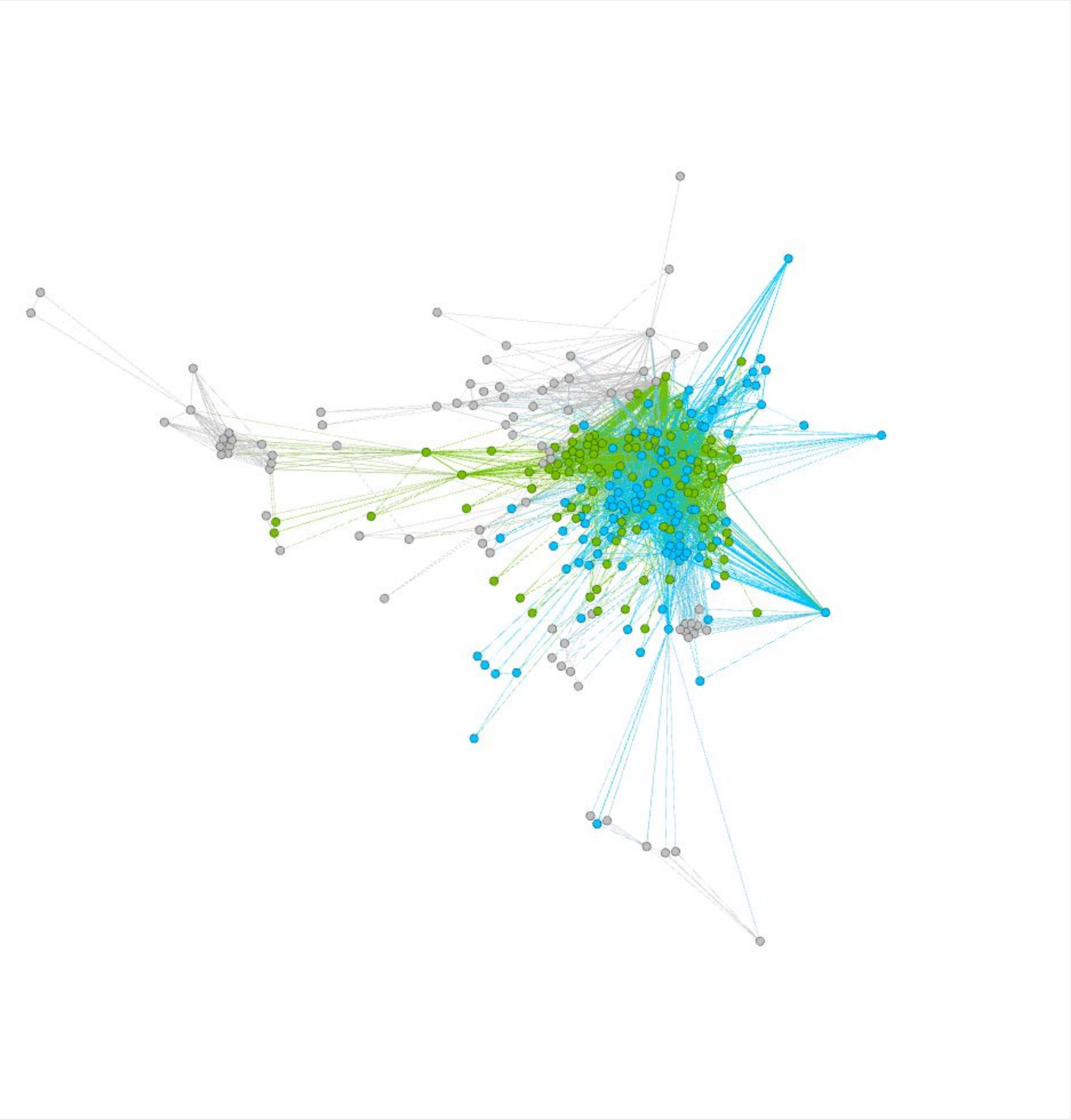
Conclusions.

This methodology report presents a network analysis driven approach to developing an alternative skills taxonomy. Such an approach could have multiple benefits to the end user including: a) identifying skills in a cluster to skill up in for better job prospects within an industry, b) identifying skills to develop in a new industry should an individual want to pivot careers and c) investigating skills relevant to multiple clusters. While there are several areas of improvements to be made, the methodology report provides a sound basis for an alternative skills taxonomy.

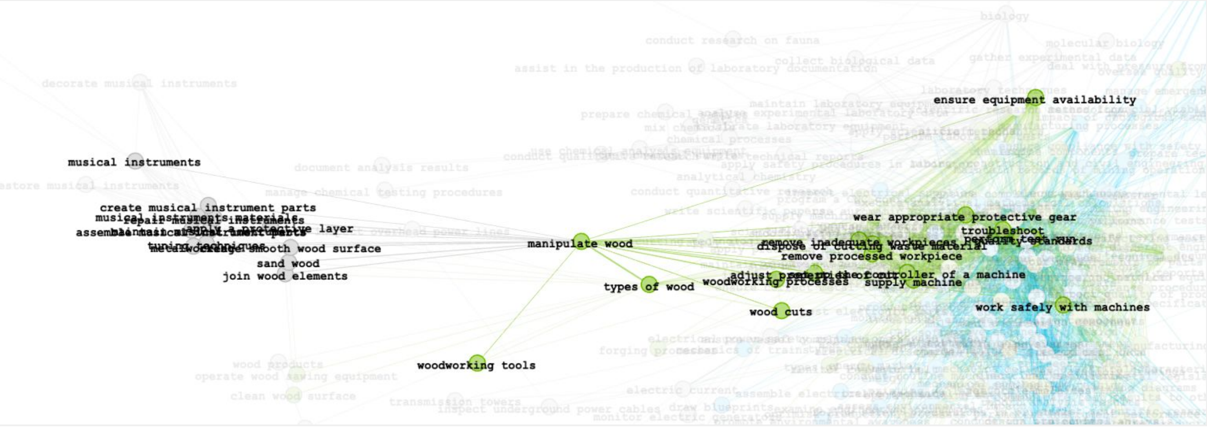
Gephi Network screenshots.



The network overall



Community 1: equipment | engineering | use | electrical | test



Community 1: focused state of two sub communities

Sources referenced.

- https://ec.europa.eu/esco/portal/document/en/87a9f66a-1830-4c93-94f0-5daa5e00507e
- https://escoe-website.s3.amazonaws.com/wp-content/uploads/2020/07/13161304/ESCoE-DP-2018-13.pdf
- https://medium.com/@adityagandhi.7/network-analysis-and-community-structure-for-market-surveillance-using-python-networkx-65413e7b7fee