ISSN 2007-9737

# Optimal Clustering of Central Bank Role Profile Descriptions

Aidan Wade[1,*], Markus Hofmann[2]

[1] Central Bank of Ireland, Data Operations,
Ireland

[2] TU Dublin, Department of Informatics,
Ireland

aidan.wade@gmail.com, markus.hofmann@tudublin.ie

**Abstract.** The Central Bank of Ireland has a set of role profiles used when recruiting new staff but which also contain information about the current skill levels in the bank and which could support project planning. The roles are manually created according to a semi-structured template and the volume of roles makes them increasingly hard to manage, requiring an NLP solution for finding similar roles and applying an appropriate grouping. Different pre-processing and dimension reduction methods are tested using K-Means and Agglomerative Clustering (HAC) with clustering metrics Davies-Bouldin and Silhouette. This suggests an optimal number of clusters in the range 70 to 130 but the correct value is subjective and requires subject matter expertise.

**Keywords.** NLP, clustering, K-means, agglomerative clustering, role descriptions.

## 1 Introduction

The Central Bank of Ireland defines a long form role profile for all positions in the bank. All staff in the bank should have an associated role profile based on their position which describes the role purpose, the role accountabilities, required skills and experience. This profile is shared with candidates as part of the recruitment process and they are updated periodically as the requirements and titles of the roles evolve.

This has led to a proliferation of role definitions, up to 782 different roles or about one for every three staff. This makes it hard to find the most apt profile for a new role and can lead to duplication of effort and further growth of the role corpus.

### 1.1 Business Objectives

The HR team that administers the roles have identified several potential improvements but the number of role documents, each between three and five pages long, would be too time-consuming for the team to complete a full manual review. Instead, machine learning options are investigated in order to:

- Identify existing, similar roles that could be consolidated (however there is no predefined classes or hierarchy).
- Extract key meta data, in particular, the role review date to a database - but hide the name of the document approver.

Following background research (Section 2), A CRISP-DM methodology was applied in order explore the existing role profile data (Section 3), develop and optimise a bespoke pre-processing routine (Section 4) and test options for clustering (Section 5). The results are discussed in Section 6 and Section 7 concludes the paper.

doi: 10.13053/CyS-28-4-5302

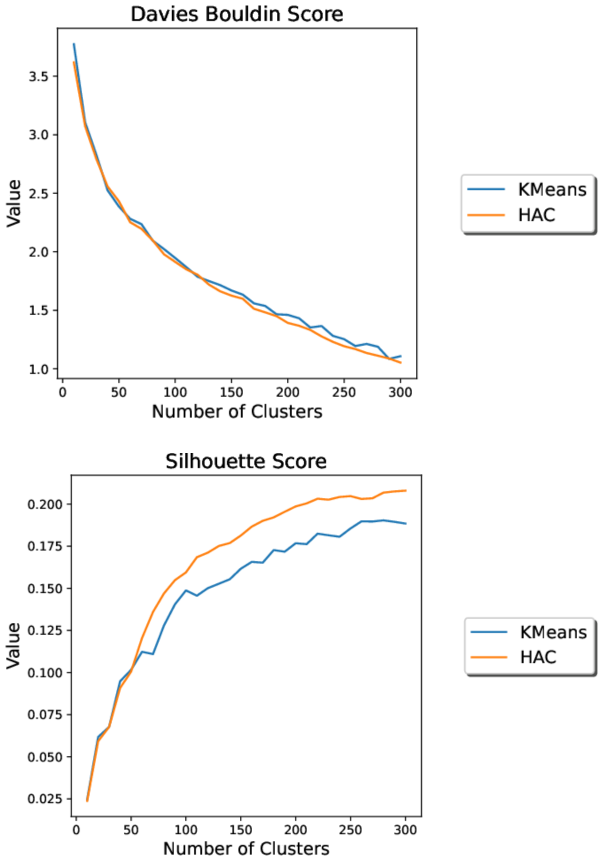


**Fig. 1.** Comparison of baseline K-Means and HAC clustering performance for a range of K values, 10-300, for 2 different clustering metrics

## 2 Background and Related Work

Previous research in this field has focused on published job descriptions on 3rd party recruitment websites. The goal is generally to cluster job profiles as preparation for potential mapping of applicant resumes and graduate profiles to open positions.

[3] uses K-Means to group almost 15k job vacancies within the relatively narrow area of “Big Data” positions within the Chinese job market. They use Sum of Squared Error (SSE) to measure clustering performance of Chinese language text leading to selection of 10 clusters within the Big Data field.

[8] also uses K-Means on a smaller sample set of 55 Indonesian language job profiles that yield a TF-IDF matrix with a vocabulary of 730 terms - preprocessing includes common techniques such as stop word removal and stemming. Silhouette coefficient is used in combination with SSE to evaluate K-Means for different values of K - using the elbow method to select an optimal value.

Hierarchical Agglomerative Clustering (HAC) can be used without prior knowledge of the optimal K value but does not result in definitive cluster list without “flattening” - choosing a distance or K value at which to stop further agglomeration. Where HAC is used in role clustering, it is typically with the goal of clustering extracted terms in a semantic sense rather than documents [10, 1].

The data used in this paper, in contrast to referenced research covers a broader range of roles, including a mixture of technical, legal, financial, research and administrative positions and these are further distributed vertically within the organisation, from entry level to senior management.

## 3 Data

### 3.1 Data Exploration

The role definition corpus comprises 782 pdf documents distributed in a sub-folder structure loosely based on the bank’s organisational divisions - this sub-folder structure could provide some information about existing relationships but these may also reinforce existing silos in the organisation and lead to duplication, as very similar roles are defined multiple times in different parts of the bank.

For example, there many roles with the title “Data Analyst”. Each role profile follows the same semi-structured template grouped under 7 sections labelled A through G.

– **Section A** includes role meta data, such as the title, organisational association (such as division, directorate), the role it reports to and how many roles report to it. This section is presented in a table format without narrative.

- **Section B** describes the role purpose, this a very high-level summary of role, one or two sentences in natural language.
- **Section C** lists the role's main accountabilities - the tasks that anybody occupying the role will be expected to deliver. They are listed in order of importance. Each is described in 2-3 sentences and there are typically between 5 and 7 accountabilities per role.
- **Section D** is split into three subsections. The first indicates the minimum education level as a checkbox and a comma delimited list of preferred qualifications. The second describes the preferred professional experience including the seniority in previous roles in three or four sentences. The last and largest section presents required skills in two lists, technical and interpersonal, that a candidate should bring to the role. These are listed as short sentences such as "*Ability to communicate complex findings and ideas in plain language*".
- **Section E** presents a list of core competencies, such as "*Teamwork & Collaboration*" and "*Develop Self & Others*". There are 12 total and the 4 most relevant to the role should have a required competence level indicated. These are rated on a scale 1 to 4. This is recorded as a check box and will be challenging to extract.
- **Section F** is optional and allows for other information to be included that falls outside the standard template sections - for example travel requirements.
- **Section G** contains more meta data, it includes who approved the role definition, their role and when. The name of the approver, where completed is the only personal identifiable information in the role specification and the HR team have requested this is not used as part of machine learning.

Using a basic white space tokenizer to split the raw role corpus yields 5,868 distinct tokens.

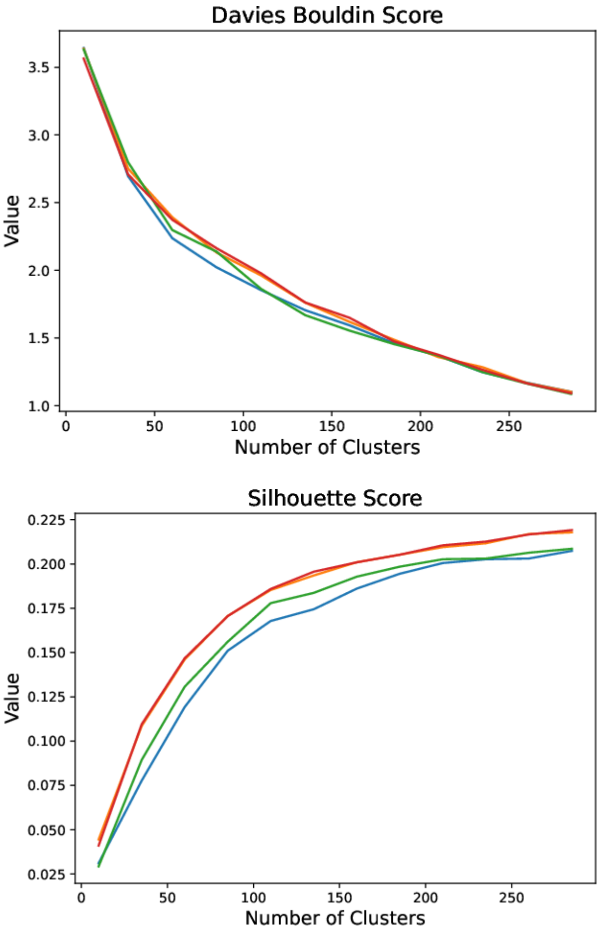


**Fig. 2.** HAC clustering performance for several combinations of stop word and phrase removal compared to the baseline (labelled 'Raw Text')

### 3.2 Data Quality

These documents are prepared for sharing with external candidates and as such, they undergo a significant review process and the spelling and grammar are excellent as a result with few errors. There are some formatting inconsistencies, for example some of the points in some sections are in the form of numbered or bulleted lists but others are separated using only a new line.

These lists do not always terminate each point in a period, many are implicitly separated by a new line character instead. When parsing the text from pdf, it's not possible to discern which new lines are due to a manually created linefeed or a single paragraph that has wrapped to a new line.

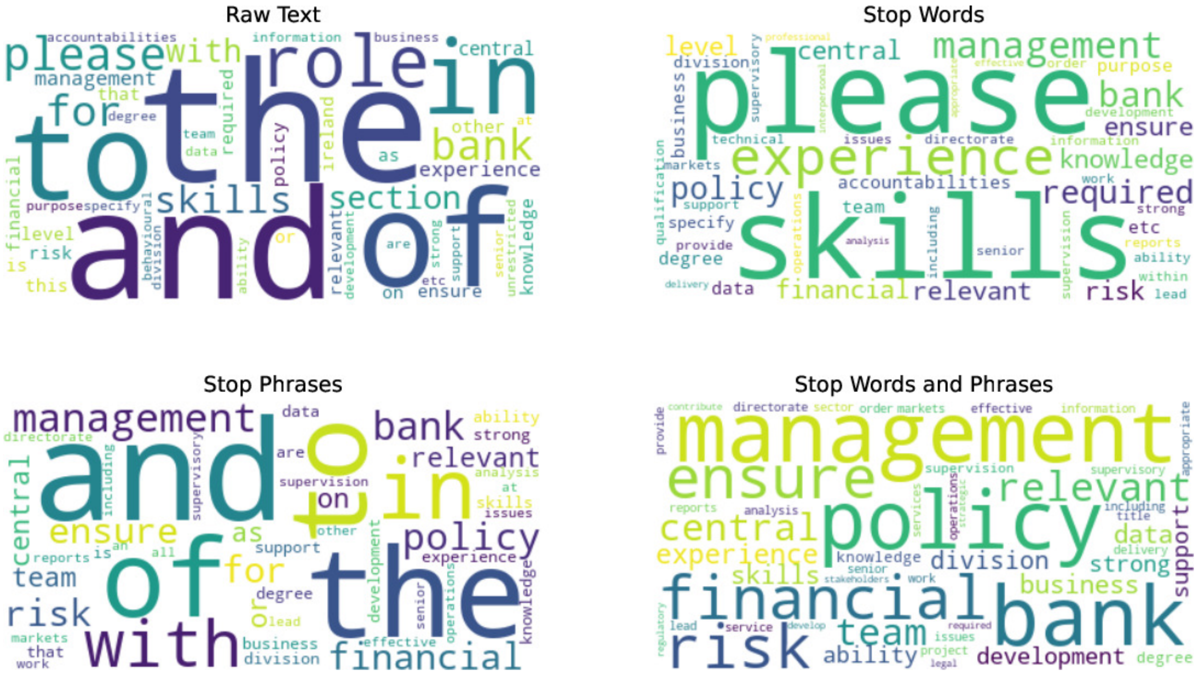


**Fig. 3.** Word Clouds for combinations of stop word and phrase removal compared to the baseline (labelled 'Raw Text')

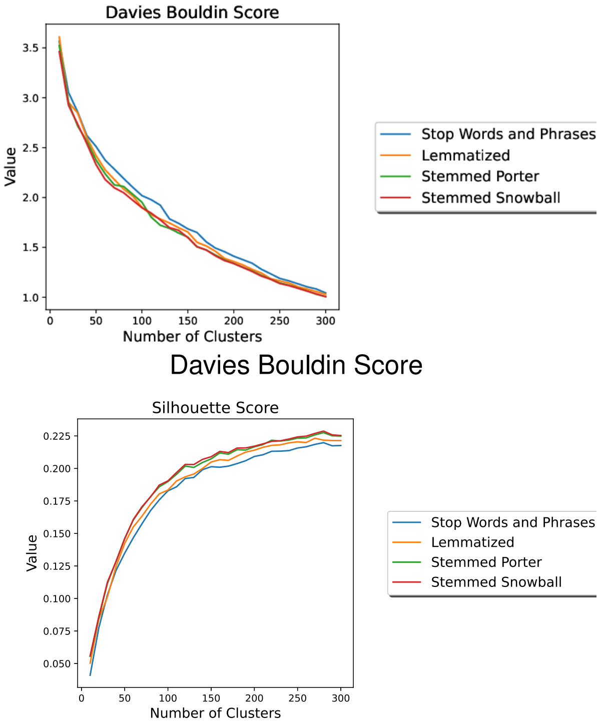
This makes it very difficult to perfectly tokenize by sentence and extract the individual skills and responsibilities into a list and a 'bag of words' representation of the relevant document section(s) is more straightforward. Because the documents follow a template, we can use the semi-structured layout to extract certain fields from specific locations across the full corpus.

Using regex patterns and the field headings, we can parse the raw text to extract these values. Although the attributes should have a limited set of acceptable values - for example the Central Bank has around 20 official directorates - as the role document is manually created, many variants of the different directorate names have emerged and in some cases the field has been incorrectly filled so there are over 70 distinct values captured in this field from the meta data in section A.

Some roles are created for use in multiple areas and these have either blank or incomplete meta data. The date of approval in Section G, where included, was written in free text and conforms to the reviewer's preferred formatting. It often excludes the day component and the month may be written in long form, a three letter abbreviation or a number. The field separator was a mix of spaces, dashes, periods and hyphens. While sections A, E and G do not contain any sentences and F is typically blank, Sections B, C and D contain a narrative style description of the role. The writing style is typical of role information (and resumés) in which the subject of the description (e.g., the candidate) is often implicit and not directly referenced.

For example, a technical skill might be "*Significant experience of design and systems architecture*" which is an incomplete sentence, grammatically - a more complete version might have started with: "*The applicant should have*". This may hinder parts of speech tagging.

Each document contains some standard text that does not vary between roles. This includes section headings and text explaining the section's function and instructions for correctly completing it. For example, Section B is prefaced by text: "*Please summarise the overall purpose of the role, i.e. why the role exists within the organisation (1-2 sentence summary of the primary accountabilities*



Davies Bouldin Score

Silhouette Score

**Fig. 4.** Impact of stemming approaches and Lemmatization on cluster metrics following removal of stop words and phrases

*as outlined in Section C below).*" These repetitive phrases do not help distinguish the document and can be removed from the text.

## 4 Data Preparation

### 4.1 Text Extraction

The text was extracted from the PDFs using PyMuPDF package for python. The text from each page was extracted and concatenated into a single string object - one per document. This field includes extraneous white space and in some cases when "printed" to pdf, it includes an automatic header with the documents sensitivity level also. A dictionary was used to define field names and a corresponding RegEx pattern which was applied to each document to extract meta data into the correct field (where available).

The python function parse() from the package dateutil was used to try and recover a date in standard format from the Section G text and some additional checks were made to check the date fell within an acceptable range.

Where the day was excluded, the first of the month was used. If a valid date couldn't be extracted, the field was left blank (113 instances). Once the date was recovered (or found unrecoverable), section G was truncated from the raw text - this section had no other value for the data mining objectives and truncation was the safest way to ensure the approver's name was removed.

The remaining text was then split by sections A through F so they could be separately processed if required during preprocessing and modelling. The text sections, along with the meta data was saved to a MS SQL database table with a single record for each document in the corpus.

### 4.2 Baseline Performance

#### 4.2.1 Clustering Metrics

In order to evaluate the impact of pre-processing, we can compare the effect it has relative to a baseline clustering performance. The data set is unlabelled and although it could be classified based on metadata such as division or folder structure, the metadata has mixed quality and does not address the main clustering objective which is to find similar roles across the existing hierarchy. [5] suggest Silhouette as an appropriate measure however other metrics suggested required prior knowledge of the target class which is lacking here.

Instead, several other standard cluster quality metrics were selected that can be used without predefined classes such as Sum of Squared Error (SSE, sum of the squared Euclidean distance from each cluster element to the cluster centre) and the Davies Bouldin (DB) score (the average of the ratio of within cluster distances to inter-cluster distance of its nearest neighbour cluster).

In the case of Silhouette, a higher score is better up to a maximum of 1 while for SSE and DB, a lower value is better. Testing showed an anomaly in the SSE metric, which increased as the dimensionality is reduced making it unsuitable

**Table 1.** Vocabulary size for different stages of processing and optional document frequency (DF) filter

| *DF Filter* | *Data Name* | *Vocabulary* |
|---|---|---|
| - | Raw Text | 5,868 |
| - | Stop Words | 5,627 |
| - | Stop Phrases | 5,716 |
| - | Stop Words and Phrases | 5,619 |
| - | Lemmatized WordNet | 4,689 |
| - | Stemmed Porter | 3,788 |
| - | Stemmed Snowball | 3,777 |
| <0.9 | Word Unigram TF-IDF | 3,757 |
| <0.9 | Word Bi-gram TF-IDF | 67,151 |
| <0.9 | Character Tri-gram | 3,929 |
| <0.9 | Character Quin-gram | 49,582 |

**Table 2.** Words removed with DF filter

| *DF Cutoff* | *Excluded Terms* | | |
|---|---|---|---|
| 0.9 | interpersonal | education | master |
| | indirect | provide | ensure |
| | diploma | technical | degree |
| | number | team | phd |
| | skill | cert | directorate |
| | management | division | knowledge |
| | experience | title | please |
| | leave | direct | specify |
| | pillar | detail | report |
| 0.8 | include | strong | support |
| | work | relevant | year |
| | development | communication | stakeholder |
| | ability | bank | level |

for assessing the effect of dimension reduction techniques. This may be because as common words are removed, it has the effect of increasing the average values in the TF-IDF matrix, increasing the magnitude of the vectors - and squared errors - as a result. For this reason, the Silhouette and Davies Bouldin scores were chosen as the primary metrics.

### 4.2.2 Preprocessing Workflow

A standard pre-processing was applied to the raw text extracted from the PDFs [4].

All characters were converted to lower case, contractions were expanded, numbers, punctuation and line breaks were removed. Multiple white space was trimmed to a single space between each word. For the baseline result, no stop words are removed or other dimension reduction techniques applied.

### 4.2.3 Vectorisation

Following this basic preprocessing a sparse document vector matrix was created by tokenizing the document by white space, converting each document in the corpus to a bag of words representation. The matrix had 782 rows (one for each document) and 5868 columns representing the full corpus vocabulary.

The word counts were then converted to a TF-IDF values where TF (Term Frequency) is a measure of how frequently a word appears in a document and IDF (Inverse Document Frequency) is measure of how rare a particular word is in the corpus. This helps to mitigate the impact of common stop words as the IDF score of these words should be low (though the scikit-learn implementation adds one to the IDF score to prevent an IDF of zero for terms that appear in every document in the corpus) [9].

### 4.2.4 Clustering Algorithms

There are multiple algorithms which can find clusters in unlabelled data. Two of the most popular are K-Means and Hierarchical Clustering (HCA) which were tested here [7, 6]. In both cases the default Euclidean distance was used initially and for HCA, the model used Hierarchical Agglomerative Clustering (HAC) based on Ward linkage. Ward linkage selects clusters to merge which will create the smallest increase in variance within the cluster - the change in Squared Standard Error (SSE) after two clusters are merged.

Both were applied using the scikit-learn package in Python. K-Means requires an expected number of clusters (the 'K' value) and HAC does not on its own return a number of clusters however using python, a range of K values can tested - in the case of HAC, the algorithm stops joining clusters once the desired K value is reached.

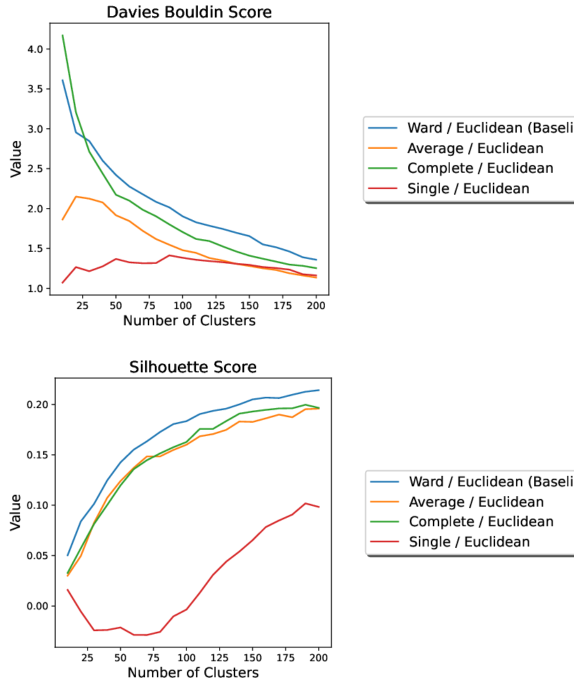


**Fig. 5.** Effect of different linkage methods on clustering metrics after lemmatization for K = 10 to K = 200 (Euclidean distance)

The results of the baseline performance test (Figure 1) do not provide a clear indication of an optimal k value. K-Means and HAC, when compared on DB score, are very similar but the silhouette score for HAC shows more improvement for $K > 60$.

Visual inspection of the curves suggests diminishing returns for K above approx. 100 but overall the selected metrics suggest the clusters are not sharply defined and subject matter expertise will be required to make a final decision on role profiles in border regions.

## 4.3 Feature Selection Tests

As typical with text documents, many of the most frequently used words do not on their own add any meaning, these are stop words such as ‘the’, ‘of’, ‘in’ etc. There are many standard dictionaries of these which allow the text to be filtered to exclude these low value terms.

The standard English stop word list included in the python NLTK package was used [2]. Additionally, data exploration indicated that there were corpus specific frequently used terms that are repeated in every role document as part of template instructions and will not add any value - e.g., “stop phrases”. This includes all of section E (the behavioural competences) because this is based on checked boxes that were not captured as part of the text extraction.

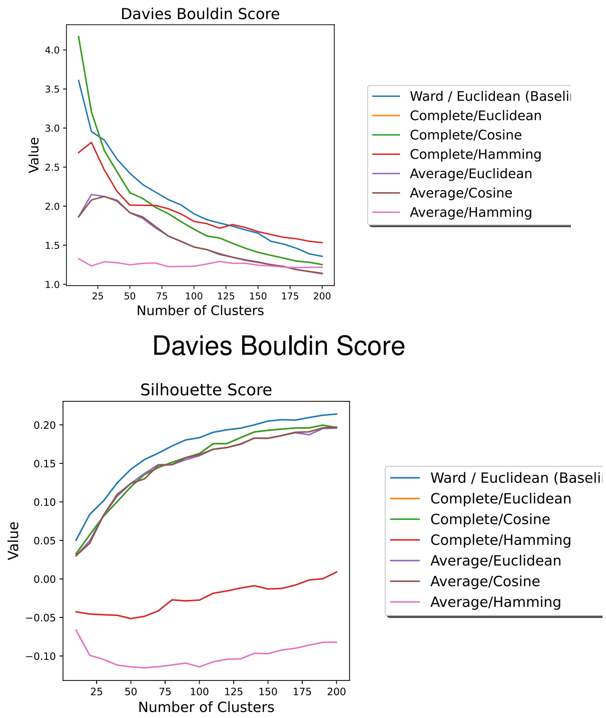
### 4.3.1 Stop Words and Phrases

The same basic preprocessing and HAC algorithm was tested with either or both of the stop words and stop phrases removed as compared to the baseline already established. The results are shown in Figure 2. Compared to the baseline result (labelled ‘Raw Text’), removing the common words and/or phrases has a positive impact with the best result is observed when both are removed according to the DB and Silhouette score.

The Word Clouds in Figure 3 are calculated from the full corpus, based on the sum of each term’s TF-IDF value from all documents to get a shorthand approximation of the most relevant terms. The ‘Raw Text’ cloud and the ‘Stop Phrases’ Cloud are made up mostly of common stop words due to their high frequency which dominate despite a low IDF value. In the ‘Stop Words’ cloud, words such as ‘please’ and ‘skills’ which are used in the static section headings dominate. When both stop words and phrases are removed, a more meaningful selection of words begins to emerge.

In addition to the dictionary-based methods, using lists of stop words and phrases, a method based on document frequency (DF) was also tested. The scikit-learn transformer, CountVectorizer, which creates the initial term frequency document matrix can optionally filter words that have very a high DF value.

Cutoff values at 80% and 90% were tested (e.g., words appearing in more than 80% and 90% of documents were excluded) and the results closely matched the result achieved with a dictionary approach with a cut-off at 90% - reducing the tolerance to 80% did not appreciably improve the metrics further.



Davies Bouldin Score

Silhouette Score

**Fig. 6.** Effect of different linkage methods and distance measurement on clustering metrics after lemmatization for K = 10 to K = 200

While for the purposes of subsequent tests the dictionary-based approach was retained, this required additional work to configure and if the role format and template changes in the future, the dictionaries would need to be updated. A method based on DF could be more robust.

### 4.3.2 Lemmatization and Stemming

The dimensionality can be further reduced by lemmatization or stemming. Stemming uses rules to truncate the endings of words in order to reduce different forms to a common base - even though that base might not be a valid word. By contrast, lemmatization seeks to convert words to a correct, common form and as such leaves the text in a more readable format though with a potentially higher vocabulary compared to stemming.

Two stemming algorithms were tested, the Porter stemmer and Snowball stemmer as implemented in the python NLTK library. A lemmatization process based on NLTK WordNet was tested also. This required first tagging each word to indicate whether it's a noun, verb, adjective etc. which is performed after stop phrases are removed but before stop words are removed to give the Parts of Speech (POS) tagger more context to distinguish between words that could be a noun or a verb.

This is in contrast to the suggested workflow in in [4] which places lemmatization after stop word removal but there is no reported indication that stop word removal would improve the quality of lemmatization - or that lemmatization would negatively impact a subsequent stop word removal process.

The results of the tests of the three algorithms are compared to the 'Stop Words and Phrases' text from Section 4.3.1 in Figure 4. The Silhouette and Davies Bouldin scores indicates that there is some small improvement observed, with the stemmed data sets showing the greatest uplift. For the purposes of these tests, the output from lemmatization is used for its greater readability.

## 4.4 Alternative Vectors and Tokenization

In addition to testing dimension reduction techniques, two alternative modes of document vectorisation were tested - using terms counts without converting to TF-IDF and a binary flag which indicated a word was or was not present in the text. The same basic preprocessing steps including stop words and phrases removal were applied without yielding improvement in results.

Although most testing was conducted on the basis of tokenization of each word based on white space (e.g., a word unigram), word bigrams (e.g., pairs of words) and character n-grams of length 3 and 5 were tested as alternatives. Again, the same basic preprocessing steps were applied and further, to account for the emergence of very common word and character groups, a document term frequency filter was applied also (DF $<$0.9). As expected, using bi-grams and character quin-grams greatly increases the dimensionality

**Table 3.** Cluster size statistics for different linkage choices and different values of K (total clusters)

| ***Configuration (Linkage / Distance Metric)*** | ***K*** | ***Max*** | ***Min*** | ***Avg*** | ***Std.Dev*** |
|---|---|---|---|---|---|
| HAC Ward / Euclidean (Baseline) | 70 | 28 | 3 | 11.17 | 6.14 |
| HAC Ward / Euclidean (Baseline) | 90 | 24 | 2 | 8.69 | 4.32 |
| HAC Ward / Euclidean (Baseline) | 110 | 17 | 2 | 7.11 | 3.51 |
| HAC Ward / Euclidean (Baseline) | 130 | 16 | 2 | 6.02 | 2.87 |
| HAC Average / Euclidean | 70 | 75 | 1 | 11.17 | 13.19 |
| HAC Average / Euclidean | 90 | 75 | 1 | 8.69 | 10.96 |
| HAC Average / Euclidean | 110 | 65 | 1 | 7.11 | 8.89 |
| HAC Average / Euclidean | 130 | 65 | 1 | 6.02 | 7.94 |
| HAC Complete / Euclidean | 70 | 61 | 1 | 11.17 | 9.81 |
| HAC Complete / Euclidean | 90 | 56 | 1 | 8.69 | 8.07 |
| HAC Complete / Euclidean | 110 | 32 | 1 | 7.11 | 5.96 |
| HAC Complete / Euclidean | 130 | 24 | 1 | 6.02 | 4.81 |
| HAC Single / Euclidean | 70 | 594 | 1 | 11.17 | 70.41 |
| HAC Single / Euclidean | 90 | 416 | 1 | 8.69 | 44.47 |
| HAC Single / Euclidean | 110 | 327 | 1 | 7.11 | 31.95 |
| HAC Single / Euclidean | 130 | 305 | 1 | 6.02 | 27.05 |

of the data set and significantly slows down the processing time. The alternative vectors and tokenization did not yield any improvement, neither with or without further DF filtering, so TF-IDF vector matrix of word unigrams was retained and used for data modelling in Section 5. The dimensions of the different tokenization tests are listed in Table 1.

## 5 Modelling

### 5.1 Model Tuning

There are two principle parameters that can affect the result of HAC and they are closely related - the linkage type and the distance metric used. The default settings in scikit-learn's AgglomerativeClustering class use Ward linkage and Euclidean distance.

Retaining Euclidean distance metric, other linkage methods were tested in Figure 5. Complete and average linkage, which to varying degrees take account of the whole of the cluster, show a similar trend compared to Ward linkage.

The two metrics disagree on which performs best however the Silhouette score - which favours Ward linkage - appears more coherent. Single linkage, which takes account of only the nearest points in neighbouring clusters is considerably worse than the alternatives.

Cosine and Hamming distance were selected as alternative distance metrics. Cosine is a popular metric in text mining because it ignores vector scale which can vary with document length however by using TF-IDF we are normalising for text length - and the role profiles are very similar in size also. Hamming distance uses a binary representation of the vector to get a logical XOR comparison of the two vectors being measured.

As Ward linkage is only compatible with Euclidean distance (in the scikit-learn implementation), the average and complete linkage were tested for each of these alternative distance metrics alongside the Euclidean baseline in Figure 6. Again, the baseline combination of Ward linkage and Euclidean distance outperformed the alternatives.

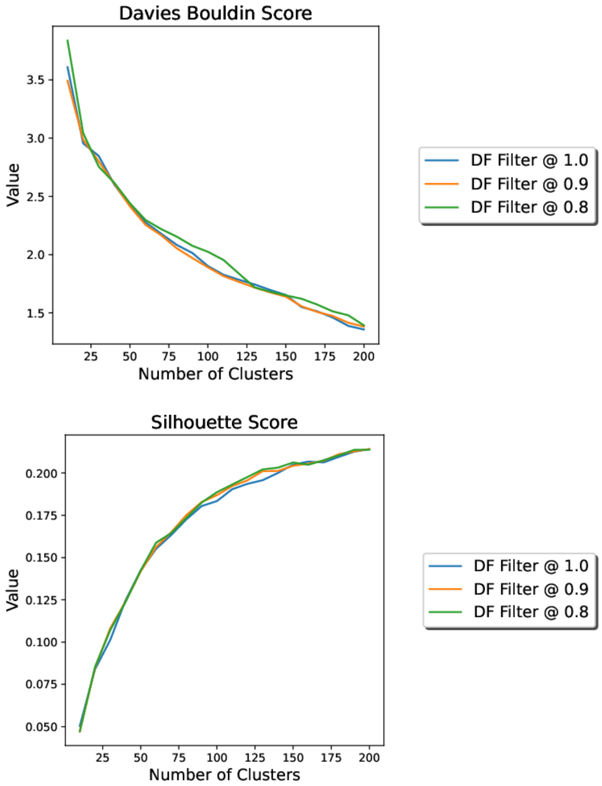


**Fig. 7.** Retesting DF filter options using final HAC configuration: ward linkage, Euclidean distance following lemmatization and stop word removal for K = 10 to 200

Hamming distance appears to be a poor choice however Cosine distance, with either complete or average linkage performs similarly to the baseline.

## 5.2 Optimal K Value

The true number of clusters ‘K’ is unknown and likely to be subjective. There appears to be an optimal range of possible cluster numbers between approx. 70 and 130 above which there the rate of metric improvement slows. In order to better evaluate whether the clusters ‘make sense’ at this scale, the distributions of cluster sizes are examined for four different values of K - 70, 90, 110, and 130. The distribution was calculated for the same combinations of linkage and distance metrics above, the baseline (Ward linkage, Euclidean) and single linkage / Euclidean - are compared in Figure 8 while cluster size statistics for the difference linkage options are presented in Table 3. The poor metric performance of single linkage can be explained by the distribution in Figure 8b. In this configuration, HAC is seen to be adding singleton clusters one by one to a super cluster creating a skewed distribution for all K values.

By contrast, the baseline configuration in Figure 8a has created a much more normal distribution without any super clusters or large set of singletons. This is likely due to Ward linkage, which encourages compact clusters that minimise the variance.

Based on these, the default HAC parameters were retained (e.g., Euclidean distance and Ward linkage) and the dendrogram in Figure 9 was created. With 782 documents, it’s too large and complex to be easily analysed graphically however the relatively consistent cluster sizes at each level can be observed quite clearly.

In order to look more closely at which clusters are merged, with what documents and at what distance, the data underlying the dendrogram was converted to a text file which detailed the HAC operation at each stage. The 782 documents initially constitute 782 singleton clusters and at each merge in the HAC process, a new cluster is created, labelled incrementally and the constituent documents are recorded.

The HAC merge steps were analysed at the K = 70, 90, 110 and 130 stage. To get a qualitative sense of the level of similarity between clusters at these points, the two child cluster sets were examined in terms of the sum and average of their component document vectors. The word clouds (Figure 10) were created using the sum of the TF-IDF values for each term while the centroid - effectively the average of the component document vectors - was used to find the terms with the highest TF-IDF values (Table 4).

There are a number of words observed in common between the two child clusters for each join which can help to make sense of the merge decision - there are also a number of words which could potentially be added to the stop words list such as *‘central’, ‘bank’* but these could also help to distinguish particular skills or experience.

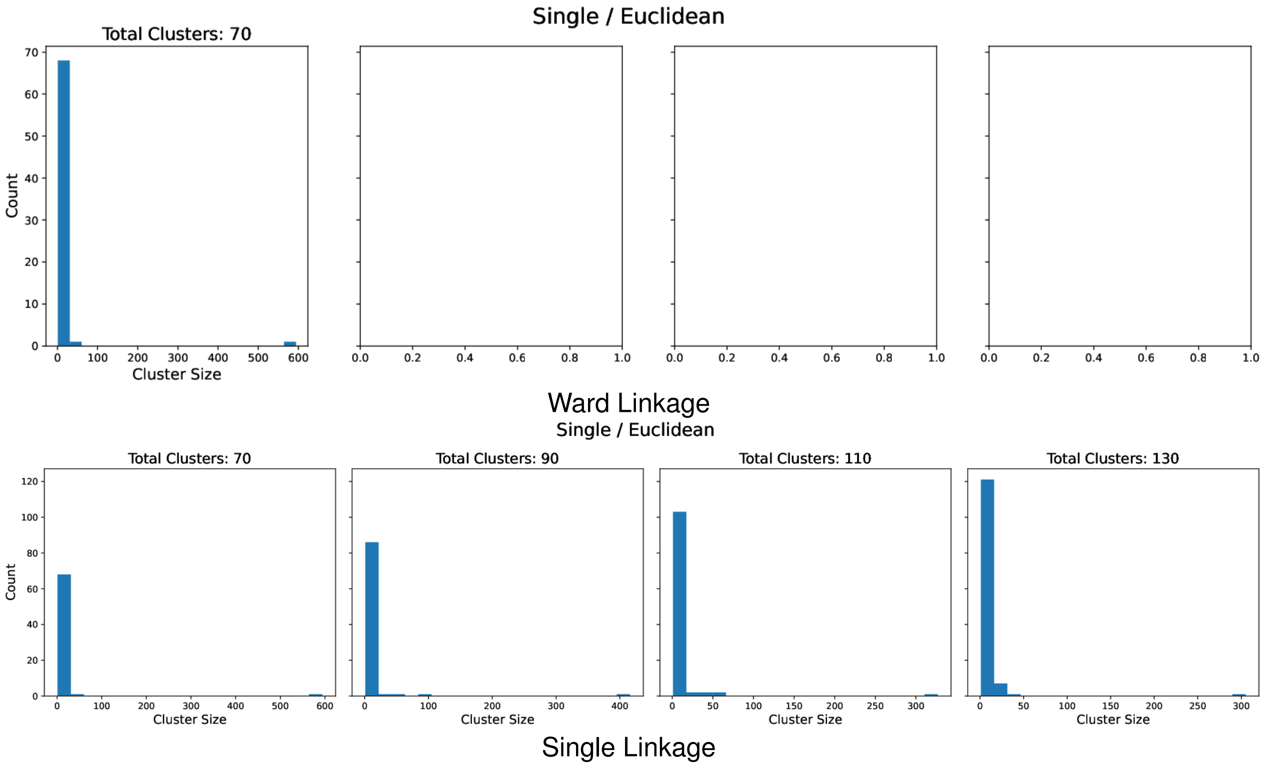


**Fig. 8.** Distribution of cluster sizes for baseline (Ward) and an alternate HAC configuration using single linkage (Euclidean distance) for 4 possible K values

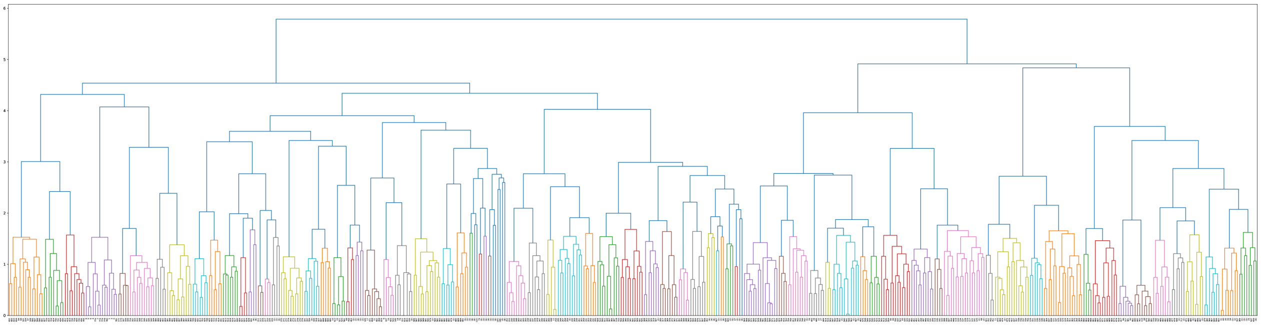
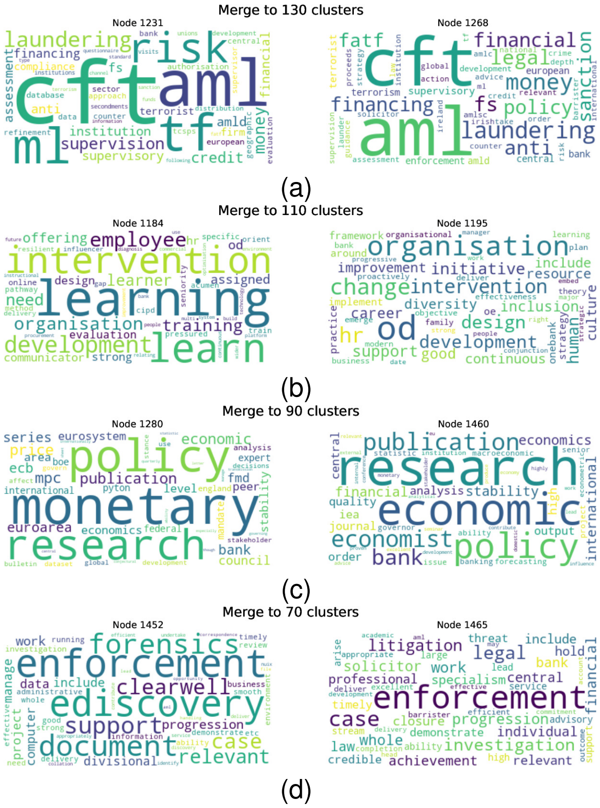


**Fig. 9.** HAC dendrogram using Ward linkage, Euclidean distance. Colour is specified based on a distance cut off of 1.683 - which would create 100 clusters. Plot is restricted to first 10 branches from top so some elements at base may represent multiple documents. Overall, the dendrogram is too large to directly interpret

To determine how common these words are, the DF filter was retested using a DF threshold of 0.8 and 0.9. As before, this had limited effect on the metrics (Figure 7). Comparing the vocabulary for the different DF filtered matrices reveals the words with a frequency above the cutoff (Table 2). The words removed with DF $>$ 0.9 include many words that are part of the template which could be removed such as *‘please’* and *‘specify’*. The word *‘bank’* has a DF between 0.8 and 0.9 but the other words in this range, though common, are more descriptive and should probably be retained.



(a)

(b)

(c)

(d)

**Fig. 10.** Word clouds for child clusters merged to leave 130 (A), 110 (B), 90 (C) and 70 (D) clusters

Based on this, a DF filter with cutoff at 0.9 was added to the preprocessing step - this is applied when calculating the vocabulary for the TF-IDF document matrix, after the lemmatization and basic preprocessing.

### 5.2.1 Detailed Analysis of Merge Decisions

The HAC process was rerun and the qualitative analysis using word clouds (Figure 10) and centroid terms (Table 4) was repeated for insight into possible valid values of K.

– K = 130: Merged clusters 1231 and 1268, each containing 7 mid to senior roles, all within the anti-money laundering (AML) area. Cluster 1268 appeared to have a slightly more legal domain focus and included more senior staff roles (heads of function and division) while cluster 1231 had a supervisory focus.

– K = 110: Merged two small clusters, 1184 and 1195, each with three roles and all within the HR department with a focus on training and organisational development. Outwardly, these clusters appear very similar, cluster 1195 may have more of a strategic focus on longer term planning and requirements.

– K = 90: Merged a smaller cluster 1280 with 4 documents into a larger set of twenty (1460). The roles include senior economists, advisers and specialists as well as senior management positions in economic departments. There is a strong common theme of research and monetary policy.

– K = 70: Clusters 1452 and 1465 comprise roles in the enforcement division and cluster 1465 particularly includes many legal roles. Cluster 1452, although it includes enforcement roles also includes several more general project and business support positions within enforcement team.

## 6 Evaluation

The selected metrics provided a view about the relative improvements and changes with different configurations but not a definitive answer on the correct number of clusters. This is likely a reflection of the subjective nature of the decisions.

The focus on this project was trying to find an optimal K value from a metric perspective which resulted in looking closely at range 70 to 130. Though this is a lot of clusters, in this context it may be unrealistically small as the Central Bank is a large organisation with a diverse range of functions and responsibilities.

An advantage of the HAC approach is being able to analyse the cluster merges one by one and allow a subject matter expert (SME) to make a final call on when to stop merging. Many of the initial merge operations from singletons to document pairs are good indicators of likely duplication and opportunities for consolidation. Further merges

**Table 4.** Top 10 terms (by TF-IDF) of child cluster centroids merged by HAC to leave 130, 110, 90 and 70 clusters

| **130 Clusters** | | **110 Clusters** | |
|---|---|---|---|
| *C1231 Centroid* | *C1268 Centroid* | *C1184 Centroid* | *C1195 Centroid* |
| cft | aml | learning | od |
| aml | cft | learn | organisation |
| tf | laundering | intervention | change |
| ml | money | development | hr |
| risk | anti | employee | intervention |
| laundering | fs | organisation | design |
| supervision | legal | training | development |
| money | financing | need | initiative |
| credit | policy | learner | support |
| **90 Clusters** | | **70 Clusters** | |
| *C1280 Centroid* | *C1460 Centroid* | *C1452 Centroid* | *C1465 Centroid* |
| monetary | research | enforcement | enforcement |
| policy | economic | ediscovery | case |
| research | policy | document | legal |
| publication | publication | forensics | investigation |
| price | economist | support | litigation |
| mpc | bank | clearwell | progression |
| economic | stability | case | work |
| euroarea | financial | relevant | whole |
| bank | economics | progression | solicitor |
| ecb | international | work | financial |

into larger clusters indicate role families which though not consolidated into a single role, can be managed more easily as a group - if a requirement needs to be updated for one, for example, it likely needs to be updated for all the roles in the family. One of the design questions to be answered by the SME would be the relevance of seniority. In some cases, middle management positions from different areas are merged first while in other parts of the business, management positions are merged initially with their subordinate roles.

## 7 Conclusion and Next Steps

There are several ways the results could be improved with further work. A vectorisation approach that took account of the order of words could help to differentiate between different abilities that use the same vocabulary. ‘Supervisor’ can refer to a management role within the bank for example but also to the supervision of regulated entities. There are many regulation and banking specific abbreviations and acronyms that might not be recognised by standard dictionaries so building a custom dictionary of these could help to consolidate terms so they have the correct weight.

The meta data was not used because its quality was mixed. If the values could be mapped to official master data lists for organisational units such as division and directorate, this could be used as an additional dimension.

It should also be possible to establish the grade (seniority) of the roles, which could help to differentiate between management and subordinate roles so these are merged consistently in accordance with SME guidance. There are options for further projects to build on the clustering work in this project. The next priority for the HR team is to extract skills from the roles in order to build an internal CV for staff based on the positions they have occupied during their career. If appropriate clusters are identified, then a classifier can be trained based on these clusters to classify new roles. This can help to avoid duplication in future.

## References

1. **Bafna, P., Shirwaikar, S., Pramod, D. (2019).** Task recommender system using semantic clustering to identify the right personnel. VINE Journal of Information and Knowledge Management Systems, Vol. 49, No. 2, pp. 181–199. DOI: 10.1108/vjikms-08-2018-0068.
2. **Bird, S., Klein, E., Loper, E. (2009).** Natural language processing with Python: Analyzing text with the natural language toolkit. O'Reilly.
3. **Debao, D., Yinxia, M., Min, Z. (2021).** Analysis of big data job requirements based on k-means text clustering in China. PLOS ONE, Vol. 16, No. 8, pp. 1–14. DOI: 10.1371/journal.pone.0255419.
4. **Hickman, L., Thapa, S., Tay, L., Cao, M., Srinivasan, P. (2022).** Text preprocessing for text mining in organizational research: Review and recommendations. Organizational Research Methods, Vol. 25, No. 1, pp. 114–146. DOI: 10.1177/1094428120971683.
5. **Jacksi, K., Ibrahim, R. K., Zeebaree, S. R. M., Zebari, R. R., Sadeeq, M. A. M. (2020).** Clustering documents based on semantic similarity using HAC and k-Mean algorithms. 2020 International Conference on Advanced Science and Engineering, pp. 205–210. DOI: 10.1109/ICOASE51841.2020.9436570.
6. **Kwale, F. M. (2013).** A critical review of k means text clustering algorithms. International Journal of Advanced Research in Computer Science, Vol. 4, No. 9, pp. 27–34.
7. **Liu, F., Xiong, L. (2011).** Survey on text clustering algorithm -research present situation of text clustering algorithm. 2011 IEEE 2nd International Conference on Software Engineering and Service Science, pp. 196–199. DOI: 10.1109/ICSESS.2011.5982288.
8. **Megasari, R., Piantari, E., Nugraha, R. (2020).** Graduates profile mapping based on job vacancy information clustering. 2020 6th International Conference on Science in Information Technology, pp. 35–39. DOI: 10.1109/ICSITech49800.2020.9392067.
9. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011).** Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, Vol. 12, pp. 2825–2830.
10. **Siswipraptini, P. C., Warnars, H. L. H. S., Ramadhan, A., Budiharto, W. (2023).** Information technology job profile using average-linkage hierarchical clustering analysis. IEEE Access, Vol. 11, pp. 94647–94663. DOI: 10.1109/ACCESS.2023.3311203.

*Article received on 19/04/2024; accepted on 31/07/2024.*
**Corresponding author is Aidan Wade.*