IDENTIFYING KEY SKILLS IN JOB MARKETS THROUGH CLUSTERING ANALYSIS

Ting-Hui Cheng (s232855), Tomasz Truszkowski (s223219)
Lukas Rasocha (s233498), Henrietta Domokos (s233107)

Technical University of Denmark (DTU)

ABSTRACT

When navigating through multitudes of job advertisements that vary in content and format, it can be an overwhelming process to understand what exactly is expected of you. In this project, we introduce a framework to analyze and understand the evolving job market by clustering analysis of job descriptions taken from LinkedIn platform. Using various methods including TF-IDF, Word2Vec, Doc2Vec, similarity-based vector representation and clustering algorithms, we grouped similar job descriptions together and extracted the most prominent skills that appear in each of the individual clusters. That has provided an insight into the specific skills demanded by various job industries. Our analysis further showed, that clustering based on TF-IDF that focuses strictly on the nouns of job posts performs the best when compared to our ”true” job clusters. This approach offers a unique perspective for understanding and navigating the job market, highlighting essential skills for career development. The code base together with a detailed overview of the project can be found in the GitHub repository.[1]

1. INTRODUCTION

In the large amounts of job posts, it can be an overwhelming process for employment seekers to find jobs that best match their skill set. The challenge lies not only in the volume of available jobs but also in understanding the diverse and often complex skill requirements specified in different job descriptions. To address this issue, our project introduces a framework built to analyse and decode the current job market through clustering analysis of job descriptions scraped from LinkedIn.

Our method involves various techniques to transform the long job descriptions into meaningful numerical representations. This transformation is then used to apply known clustering approaches resulting in groups of similar job posts. The main goal of our analysis was to extract the most required skills for the different groups of job descriptions, effectively helping job seekers to orient themselves around the job market by knowing which skills are in demand across different types of jobs.

2. DATA ACQUISITION AND PREPARATION

LinkedIn is a professional networking platform that plays a pivotal role in the modern job market. With approximately 50 million people using it for job hunting weekly, and around 90 job applications submitted per minute on the platform, it stands as a primary hub and is usually the number one choice when it comes to searching for new career opportunities. [1]

2.1. Data collection

To collect data that would help us analyze the current job market and build our solution, we decided to scrape public job posts from *Linkedin* directly. We built a custom scraper, which in addition to the job description also collects metadata such as: *title of the job, date posted, number of applicants, industry it belongs to (e.g. technology) , the function of the job (e.g. administrative), company name* and others. We decided to scrape *all* newly posted jobs (i.e. no keywords selected) from Denmark, Czechia, Taiwan, Poland and Hungary, as these are the countries of our origin. This collection resulted in 6570 raw datapoints.

2.2. Preprocessing

Once collected, the data required extensive preprocessing to ensure its suitability for the various methods that would be employed later. We applied several text preprocessing techniques to clean and transform the job descriptions as well as a general cleanup of the other attributes. These steps included:

- Word Filtering: Eliminate words with numbers and special characters ensuring text clarity.
- Date Standardization: From *LinkedIn* we obtained dates in the format: *posted x days ago* so the exact date needed to be inferred from the current date.
- Word Separation: From *LinkedIn*, words at the end of lines were concatenated e.g. *”requirementsYou’re”*. These words needed to be separated.

[1] Code can be found at (also with a concise outline of the project): https://github.com/lukyrasocha/02807-comp-tools

- Text Processing: This involved lowercasing, punctuation removal, tokenization, stop words removal, and lemmatization, focusing on extracting meaningful text.
- Data Cleaning: Removed duplicates, filtered out non-English entries, standardized categorical data, and cleaned formatting inconsistencies.
- Finalization: Post-processing involved removing overly short descriptions (less than 3 words) and saving the cleaned dataset in a structured CSV format for analysis.

This resulted in 1865 clean datapoints.

2.3. Ground Truth Establishment

To be able to adequately evaluate the various clustering methods used in the project, we needed to find the ”ground truth”, i.e. establish the ”true” labels for each job offer. To do this, we experimented with different methods to categorize jobs, each of which revealed unique insights and challenges.

2.3.1. One-Hot-Encoding

As a starting point, based on all values present in the jobs' attributes *function* and *industries* we have defined general job categories, such as 'Management and Leadership' and 'Technology and Information'. This enabled us to map each job listing to a broader category. From there we employed one-hot-encoding to convert these categories into a binary format as shown in table 1. Encoded in this way, we tried to cluster the values using the *k-means* method. This approach, however, had limitations: it couldn't capture all aspects of a job offer, and relying solely on 'industry' and 'function' labels was insufficient for a comprehensive understanding of the job market.

Table 1. Sample of one-hot-encoded dataset

id	title	.....	Healthcare & Science	Education & Training
3666672537	clinical data analyst	.....	1	0
3726117997	instructional designer	.....	0	1

2.3.2. Keywords-Based Categorization

Moving beyond one-hot-encoding, we identified 20 general job categories and assigned relevant keywords to each, e.g.

```
keywords = {
  Software & IT: ['software', 'it' ...
  ...
}
```

Job offers were then categorized based on keyword occurrence in their descriptions. While straightforward, this method was sensitive to the specific keywords used, where even minor changes, for example adding one popular word into certain category, significantly impacted the distribution of categorization.

2.3.3. Large Language Model (LLM) Categorization[2]

Finally, we tried to use undoubtedly popular LLMs. In particular we utilized *GPT3.5-turbo* model from *OpenAI* [2], to categorize job offers for us. We crafted prompts (see appendix) for the model to act as a professional recruiter categorizing job descriptions into one of 20 predefined categories. Despite occasional deviations from the rules, with the help of additional mapping, this method proved to be the most effective, accurately reflecting relevant job categories.

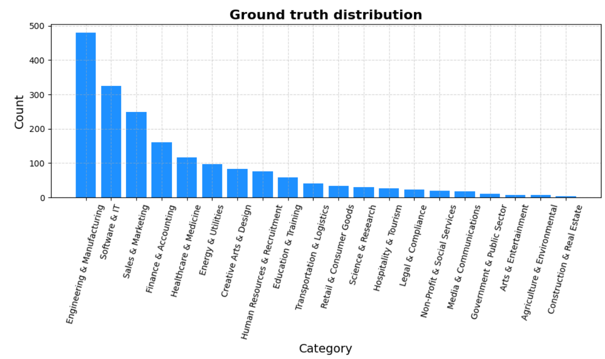

Fig. 1. Ground Truth categories, infered using *gpt-3.5-turbo*

3. METHODOLOGY

After establishing the ground truth, our focus shifted to clustering. For this, however, we had to solve one of the biggest challenges of this project - find a suitable numerical representation, which will actually help to properly represent job descriptions in a multidimensional space.

3.1. Numerical representations

Text transformed to numerical vectors to some extent should represent their content and 'meaning', and thus get relevant results with the following operations on them. To achive this, we have explored less and more advanced methods

3.1.1. TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) was the first method we explored. In this approach, we processed the cleaned job descriptions through a TF-IDF vectorizer, changing them into numerical format suitable for clustering.

3.1.2. TF-IDF variation

In addition to TF-IDF this approach includes a word type analysis were each relevant word, non linker words, within the job descriptions is classified to a verb noun or adjective.

[2] Methods marked with ”*” are the ones not taught during the course

This was performed with natural language processing (NLP) techniques, including part-of-speech tagging. The aim of word type classification is to over-represent relevant words like the values, skills and action words, within the analysis.

The verb group isolates action-oriented words, providing insights into the dynamic responsibilities associated with various roles. The noun group focuses on substantive elements, extracting key concepts that define the job positions. The adjective group extends this analysis to descriptive qualities, to discover the values sought after by employers.

*3.1.3. Word2Vec**

Word2Vec was another explored method of transforming text. Behind the scenes, *gensim's word2vec* method [3] trains a neural network model to embed words from the job descriptions into a high-dimensional vector space. After that, each job description was transformed into single vector by averaging the vectors of the words it contained. The idea is that words with similar meanings are placed close together in this space, effectively capturing their semantic relationships. Based on these relationships the k-means algorithm was used to find the corresponding clusters of job posts.

*3.1.4. Doc2Vec**

Doc2Vec expands *Word2Vec*, by adding an additional embedding for the entire document identifier (in our case, job description). During the neural network training, the model then not only learns the individual vector representations of the words, but also those of the entire documents. [4] This allows the method to capture the context of all the words within each specific job description, encapsulating its overall content and structure. The idea remains the same, the job posts are placed in a vector space, where similar posts are located closer together, establishing a meaningful spatial arrangement.

3.1.5. Similar items

Employing methodologies such as shingling, signatures, and Jaccard similarity proves beneficial in comparing textual data and identifying patterns within job descriptions. This approach can help us to estimate the likeness between text from different job, allowing us to cluster results based on these similarity values. First, we selected a hash function mapping two-word strings to a set number of shingles. Then, we utilized minhashing to generate signatures and employed Jaccard similarity to compare the signatures of each job, offering insights into their similarities. Based on the value, we determined clusters using k-means algorithm.

3.2. Clustering

Based on representations obtained by each method described in 3.1, we have used well-known *k-means* algorithm to cluster the data. Here, we in advance set the number of clusters (k) to 20, aligning with the number of general job categories previously established in the ground truth.

3.2.1. Network Graph and community detection

In addition to employ the k-means algorithm, we explored an alternative approach for job grouping using a network graph. This involved constructing a network graph with weighted edges based on the similarity between each two of job description and identifing their communities. The Louvain Community Detection Algorithm* was applied to identify communities based on modularity optimization, which is beneficial to capture intricate interrelated structures. Although acknowledging its potential weakness in deterministic optimization leading to local optima compared to the Girvan-Newman Algorithm, we still chose the Louvain method due to its efficacy in our specific scenario.

3.3. Skill Extraction

This section outlines our attempts to extract relevant skills from each cluster of job offers, utilizing two different AI models.

*3.3.1. Token-classification model**

Initially, we utilized a pre-trained token classification model from Hugging Face which was trained to extract hard and soft skills from text. [5] [6]. However, this model's performance was insufficient for our needs. It tended to extract very general terms like 'project' and 'team', which did not effectively represent the specific skills required for different job categories.

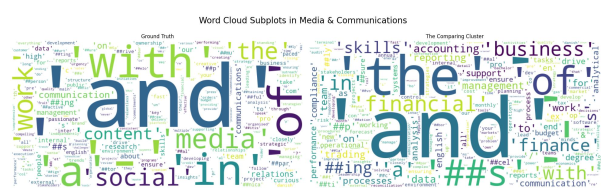

Fig. 2. An example of Hugging Face's extracted skills for the TF-IDF nouns clustering

*3.3.2. Large Language Model**

Given the limitations of the Hugging Face model, we then, again, turned to *OpenAI's GPT3.5-turbo* [2]. Using a carefully crafted prompt (see appendix), we directed the model to extract up to 10 skills from each job description without inferring or adding skills not explicitly mentioned. The performance of GPT was markedly better. It successfully captured a range of specific skills, providing a clearer and more relevant overview of the skills required for various job categories.

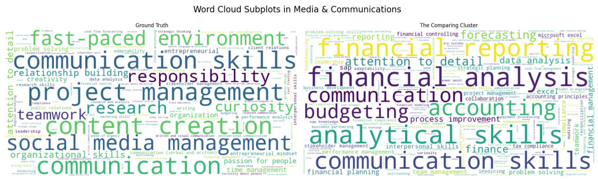

Fig. 3. An example of GPT3.5 extracted skills for the TF-IDF nouns clustering

=

4. EVALUATION AND RESULTS

This section evaluates various job clustering methods based on descriptions, using Normalized Mutual Information (*NMI*) and Rand Index (*RI*) metrics. The analysis compares results with GPT-3.5-derived categories, revealing that K-means using TF-IDF noun vectors performs the best. However, visualizations indicate some clusters overlap, demonstrating the method's effectiveness with potential for improvement in achieving finer distinctions.

4.1. Evaluation metrics

In evaluating the performance of clustering algorithms, we chose to use *NMI* and *RI* metrics. The decision to employ these two metrics due to their ability to consider partial similarity, allowing for evaluation even in situations where a one-to-one correspondence between true and predicted clusters cannot be guaranteed.

*4.1.1. Normalized Mutual Information (NMI)**

NMI [7] quantifies the mutual agreement between the ground truth clusters labels and the predicted ones. This metric is normalized by considering the entropy associated with both the true and predicted labels. The outcomes ranges from 0 to 1, where a value of 1 indicating perfect alignment with the true and predicted cluster labels.

*4.1.2. Rand Index (RI)**

RI [8] creates a confusion matrix which categorizes pairs of labels into true positive, true negative, false positive, and false negative. It is calculated as the ratio of the sum of true positives and true negatives to the total number of pairs. The resulting score ranges from -1 to 1, where 1 indicates a perfect match, 0 suggests random agreement, and -1 denotes disagreement. However, a limitation is that the metric can be sensitive to the distribution of cluster sizes.

4.2. Performance analysis

To determine the most effective clustering method, we benchmarked each approach against our established ground truth, derived from GPT-3.5 categorizations. This ground truth was chosen, as it seemed the most coherent to how we would have annotated the job descriptions ourselves. Figure 4 shows that based on the NMI score, K-means using *TF-IDF* noun vectors performed the best, suggesting that to group job descriptions to meaningful clusters, one just needs to look at the *nouns* that appear in the individual texts.

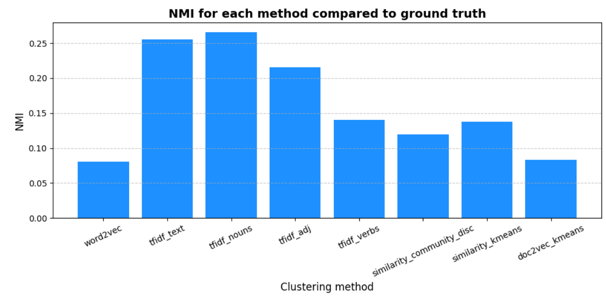

Fig. 4. *NMI* scores of clusterings compared to GPT ground truth categories

The winning clustering approach is visualized in Figure 5. This plot shows, that apart from groups $0, 4$ and 18 the clusters seem to overlap after reducing the vectors to two dimensions using principal component analysis (PCA). The illustration thus indicates, that even though the *TF-IDF* based on nouns is the best performing method, it still can't provide deep enough detail to distinguish the job descriptions into more meaningfully separated clusters. Nevertheless, the individual clusters still seem to capture similar job descriptions that differ in content from others.

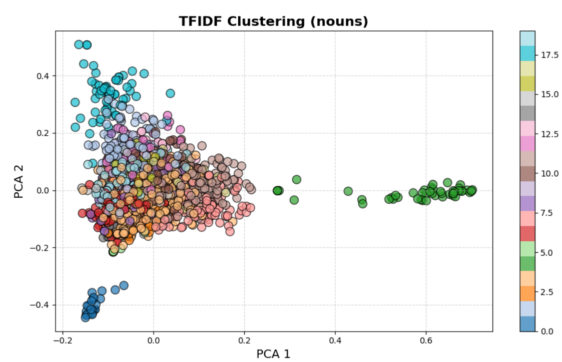

Fig. 5. K-Means clustering using TF-IDF vectors based on the nouns of the job descriptions (plot shows Principal Component 1 vs Principal Component 2)

To get a better idea on how the individual clusters look like, we provided some examples in Table 2. We can see that cluster 18 has more technical or project management related jobs, while cluster 7 appears to focus more on marketing, communications, and creative roles.

Table 2. Example of Job Titles from Clusters 18 and 7

Cluster 18	Cluster 7
Supply Chain Specialist	Marketing Project Manager
Senior Director Global Construction Procurement	Brand Communication Leader
Head of Floating Wind Technology	Social Media Specialist
Investor Relations, Finance	Junior Marketing Specialist
Head of Offshore Construction	Creative Futures is Looking for Interns
Portfolio Strategy Lead	Email Marketing Specialist
Senior Project Manager	Head of Growth
Category Manager Global Procurement	Technical Content Specialist
Quality Supervisor	Search Marketing Manager
Software Project Manager	Social Media Content Creator

5. INSIGHTS AND DISCUSSION

To better understand the key skills in each cluster, we need to integrate the results of clustering and skills extraction. This involves determining the word frequency for each cluster and visualizing the findings for analysis.

5.1. Common skills

In our analysis of job descriptions, after compiling a list of extracted skills from job descriptions, we implemented lemmatization on the words to unify word variations into a singular representation. The purpose was to ensure a consistent and accurate representation of skills, and we proceeded to quantify and sort the occurrence of each lemmatized word inside each cluster.

5.2. Visualisation

After determining word frequencies, we visualized our results in two different ways.

- Words distribution: This assists in identifying the top 5 most frequent words (Figure 6), providing a convenient way to observe the most prevalent skills required for similar jobs within a single cluster.

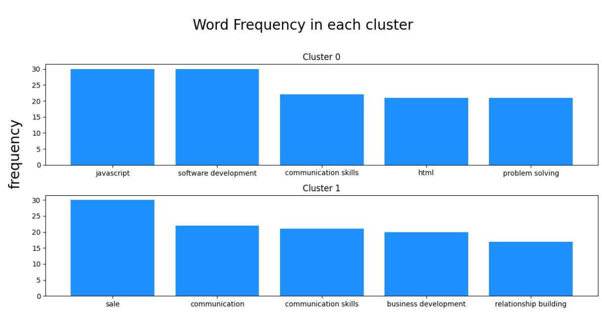

Fig. 6. Partial view of word distributions based on clustering results using TF-IDF noun vectors

- Word Cloud Visualizations: This aids in visualizing all the extracted skills and readily emphasizes key skills that require attention, providing a quick and accessible overview (Figure 7).

Fig. 7. Partial view of word cloud based on clustering results using TF-IDF noun vectors

Despite the presence of some similar terms, such as 'communication skills' and 'communication', we can identify the specific skills required for each clusters after visualization.

5.3. Comparison of skills

Once skills that occurred most frequently in each cluster have been established, we decided, for a sort of 'post-processing' comparison. The core of our analysis was to calculate cosine similarity between skill sets of each cluster considered as 'Ground Truth' and all clusters derived by certain method described in subsection 3.1. That helped us identify the most closely matching clusters. For high degree of similarity (above a threshold of 0.8), we created visual comparisons. These comparisons, in the form of word clouds or histograms, illustrated the overlap and differences between skills within corresponding clusters. One of the example has been shown in figure 8.

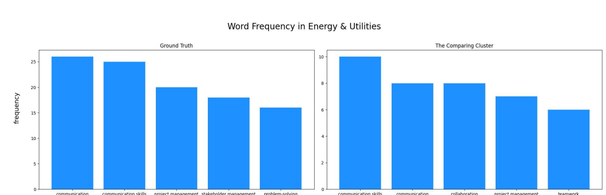

Fig. 8. Comparison of extracted skills in Energy & Utilities category and corresponding cluster.

6. CONCLUSION

In summary, this project has dealt with the dynamic landscape of the job market, with a particular focus on tackling the complexity of interpreting varied job descriptions found on *LinkedIn*.

Clustering analysis of various different representations of job descriptions have been explored, including *TF-IDF, Word2Vec, Similarity*, and *Doc2Vec*, to group job descriptions based on their similarities. Our findings show that the *TF-IDF* approach, particularly when focusing on noun vectors, was most effective in forming meaningful clusters of job descriptions. For skill extraction two different AI models were employed: a pre-trained token-classification model and *OpenAI's GPT-3.5 Turbo*. We then visualized the skills distribution within clusters using histograms and word clouds, offering clear insights into the most frequently required skills for different job categories, simplifying the process of interpreting job descriptions for job seekers.

Our method of clustering and skill extraction empowers job seekers with the knowledge to develop skills that are most relevant and in demand in their chosen fields.

7. REFERENCES

[1] Techreport, "Important linkedin statistics data & trends [2023 updated]," https://techreport.com/statistics/linkedin-statistics/, 2023, Accessed: 2023-11-27.

[2] OpenAI, "Api reference," 2023.

[3] Radim Rehurek and Petr Sojka, "Gensim–python framework for vector space modelling," *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, no. 2, 2011.

[4] Quoc V. Le and Tomas Mikolov, "Distributed representations of sentences and documents," 2014.

[5] Mike Zhang, Kristian Jensen, Sif Sonniks, and Barbara Plank, "SkillSpan: Hard and soft skill extraction from English job postings," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Seattle, United States, July 2022, pp. 4962–4984, Association for Computational Linguistics.

[6] Mike Zhang, "Huggingface, fine-tuned token classification model," https://huggingface.co/jjzha/jobbert_knowledge_extraction,https://huggingface.co/jjzha/jobbert_skill_extraction.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn, normalized mutual information," https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn, rand index," https://scikit-learn.org/stable/modules/generated/sklearn.metrics.rand_score.html.

A. CONTRIBUTION (NAME INITIALS)

A.1. Report

	TT	THC	LR	HD
Abstract	x		x	x
Introduction			x	x
Data section	x		x	
Methodology	x	x	x	x
Evaluation		x	x	
Discussion	x	x		
Conclusion				x

A.2. Implementation

	TT	THC	LR	HD
Data collection			x	
Data preprocessing	x		x	
TF-IDF	x			x
Word2vec	x			
Doc2Vec			x	
Similarity		x		
Ground truth establishment	x			
Clustering evaluation	x		x	
Skill extraction	x	x	x	
Skill analysis		x		
CLI pipeline & Jupyter notebook			x	x

B. PLOTS

To see the rest of the plots (i.e. scatter plots of all the clustering approaches, please view https://github.com/lukyrasocha/02807-comp-tools/tree/main/figures

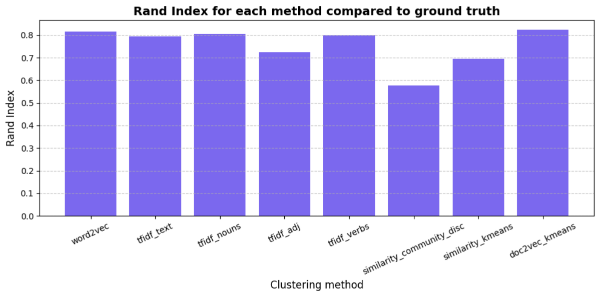

Fig. B1. Rand Index Score of clustering methods compared to GPT Ground Truth

C. LLM PROMPTS

- Prompt for ground truth extraction: *You are a professional job recruiter. Your task is to categorize a job description with keywords into one and only one of the specified 20 categories: CATEGORIES_LIST. You are not allowed to use any other categories.*
- Prompt for skill extraction: *You are an expert in job analysis. Your task is to extract at most 10 skills required for a job based on its description. Do not infer or add skills not mentioned in the description. You are required to present me the skills in a raw list format: [skill1, skill2, ... skill10].”*

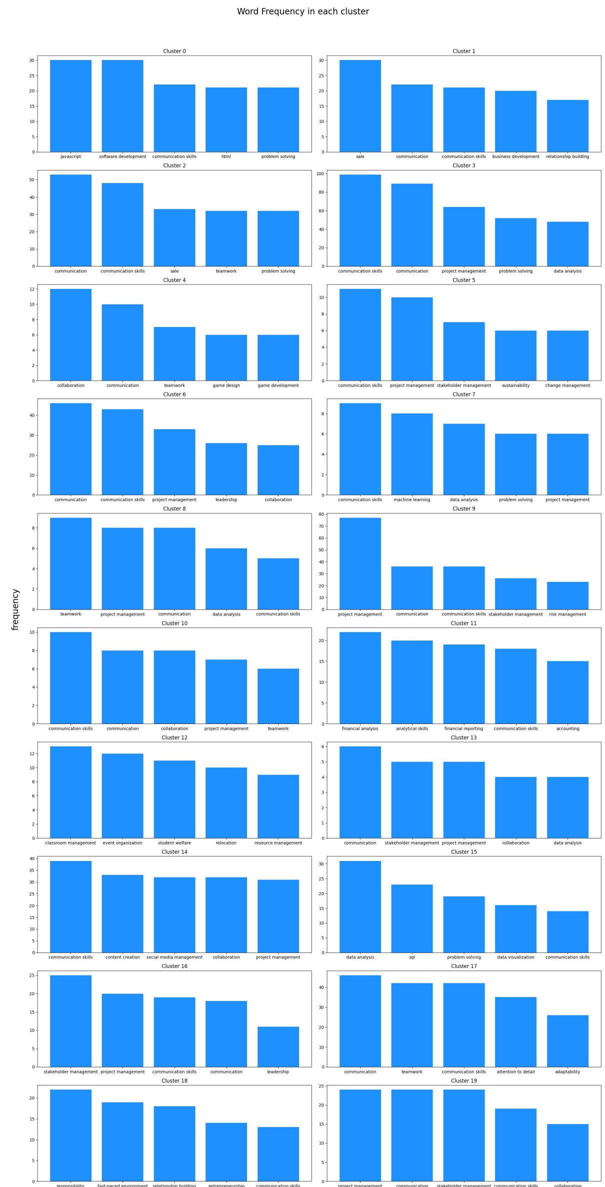

Fig. B2. Whole word distributions based on clustering results using TF-IDF noun vectors

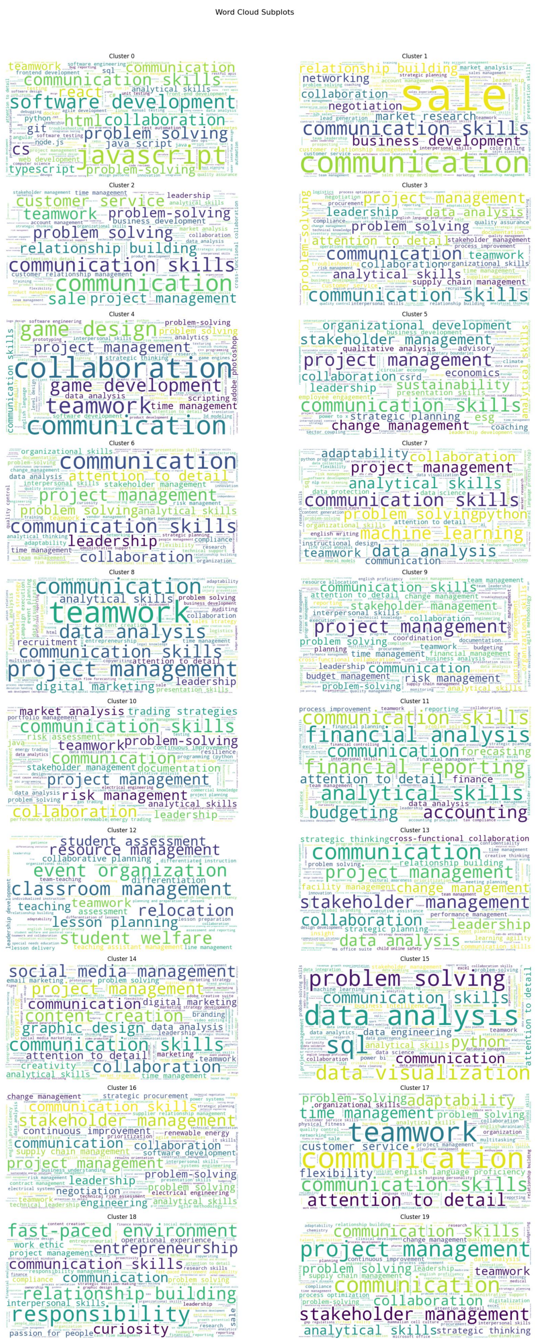

Fig. B3. Whole word clouds based on clustering results using TF-IDF noun vectors