Analyzing common themes in job roles and requirements across globally-focused humanitarian and development organizations using topic modeling

Assessment 3

Keywords: Natural Language Processing (NLP), Machine Learning, Latent Dirichlet Allocation (LDA), Non-negative matrix factorization (NMF)

I. Overview

This project sets out on a novel path by examining job roles and requirements throughout major international development and humanitarian organisations, including UNICEF, The World Bank, The United Nations Development Programme, and more. The study attempts to disentangle the complex web of shared traits, organisational priorities, and skill requirements that define these entities by applying advanced topic modelling techniques, mainly Latent Dirichlet Allocation (LDA).

This analysis, which acknowledges the growing importance of HR analytics (Margherita, 2022) in strategic decision-making, was influenced by the innovative research conducted by Thakral et al (Thakral et al., 2023). Given that these companies work in a variety of industries, including security, health, development, and food supply, it is not only insightful but also essential to develop global HR strategies to consider the similarities in the demands of their workforce. The study is expected to provide important insights into the trends in the need for particular skills and the fundamental principles that guide these organisations' goals.

Furthermore, drawing on insights from the literature, such as the work of Bansal et al. (Bansal et al., 2017) on content-based job recommendation systems, this project will enhance the understanding of job-role alignment with individual skills and preferences, further enriching the analysis. The comprehensive framework developed by Thakral et al. for HR analytics using LDA-based topic modeling serves as a foundational reference, demonstrating the potential of machine learning in transforming HRM practices.

By leveraging topic modeling, the study aims to highlight regional differences in job roles, how global trends are adapted to local contexts, and the evolving nature of skill requirements in a rapidly changing world. This approach aligns with the findings of the literature review (Ao et al., 2023; Székely & Vom Brocke, 2017; Wahid et al., 2022) that underscore the need for a structured approach to HR analytics, one that integrates statistical methods, HR roles, and emerging technologies like AI and blockchain (Ge et al., 2022). The expected outcomes of this project are crucial for stakeholders ranging from policy makers to educational institutions, offering a data-driven roadmap for aligning

educational curricula, workforce training, and recruitment strategies with the dynamic demands of these globally-focused organizations.

II. WebCrawler

This section describes the creation and operation of a web crawler designed to collect job posting data. The data is sourced from Austria's digital government agency (*Oesterreich.Gv.at – Austria's Digital Government Agency*, n.d.), with job listings specifically extracted from the Archiv des Amtsblattes zur Wiener Zeitung (*Elektronische Verlautbarungs- und Informationsplattform des Bundes (EVI)*, n.d.). The website structure is conducive to web scraping, presenting job listings in a consistent format, allowing for systematic extraction of relevant information such as employer, job title, location, closing date, publishing date, and job description. The crawler extracted data from international job postings published between July 1, 2011 and June 30, 2023.

The crawler adheres to the legal framework for data usage as outlined on the EVI platform. The platform provides centralised, free access to public information, including job listings, emphasising that the data can be accessed and used without restriction. The web crawler abides by these terms by extracting publicly available data without infringing on privacy or proprietary information.

The web crawler is written in Python and employs Selenium for dynamic webpage interaction as well as BeautifulSoup for HTML parsing. The Selenium WebDriver browses the website, visiting each job listing page. The HTML content is then parsed by BeautifulSoup to extract the required data fields. Sleep intervals are included in the crawler to mimic human browsing patterns and avoid overloading the server.

The crawler's code (Code 1) is designed to deal with common web scraping issues like dynamic content loading and pagination. To manage potential interruptions during the scraping process, error handling mechanisms are integrated, ensuring data integrity and continuity.

The data extraction process involves the following steps:

- Initialization: Launching the WebDriver and navigating to the base URL of the job listings.
- Page Navigation: Iteratively accessing each page of job listings within the specified date range.
- URL Collection: Extracting individual URLs for each job listing.
- Detail Extraction: Accessing each job listing's URL to scrape detailed information, including employer, job title, place, closing date, publishing date, and job description.

- **Data Compilation**: Aggregating the extracted data into a structured format (e.g., a Pandas DataFrame) and periodically saving it to a CSV file, as well as joining all CSV files thereafter.

The crawler is built to navigate and extract data from up to 200 job ads. Considerations for performance include minimising loading times and avoiding IP blocking or server overload. The crawler can only extract information from HTML-formatted material and cannot extract information from documents or external links. It is also limited by the structure of the website and any changes to it, necessitating modifications to the crawler's code periodically to retain operation.

The web crawler is a bespoke solution for pulling job listing data from the website of Austria's digital government agency. It is developed with legal, ethical, and technical considerations in mind, ensuring efficient and appropriate data collection.

```
#%%
# Initialize the WebDriver
driver = webdriver.Chrome()

# Scrape job details function
def scrape_job_details(job_url):
    driver.get(job_url)
    time.sleep(3)

    job_page_source = driver.page_source
    job_soup = BeautifulSoup(job_page_source, 'html.parser')

    publishing_date = job_soup.find('h1', class_='ArchivDetail__ArchivHeading-sc-
zj9bbm-4').get_text(strip=True)
    publishing_date = re.search(r'(\d{2}\.\d{2}\.\d{4})', publishing_date).group(1)

    # Extracting specific details from the job listing
    article_content = job_soup.find('article')

    # Extracting the employer
    employer = article_content.find_all('p')[0].get_text(strip=True)

    # Extracting the job title from consecutive bold elements in multiple paragraphs
    job_title = ''
    for p_tag in article_content.find_all('p')[1:]:
        bold_tags = p_tag.find_all('b')
        if bold_tags:
            if job_title:  # Add a space if job_title already has some text
                job_title += ' '
            job_title += ' '.join(tag.get_text(strip=True) for tag in bold_tags)
        else:
            break
```

```
    place = article_content.find('p',
string=re.compile('Place:')).get_text(strip=True).replace('Place:', '').strip()
    closing_date = article_content.find('p', string=re.compile('Closing
Date:')).get_text(strip=True).replace('Closing Date:', '').strip()

    job_description = ''
    for tag in article_content.find_all('p')[3:]:
        text = tag.get_text(strip=True)
        if 'Place:' not in text and 'Closing Date:' not in text:
            job_description += text + ' '

    return {
        'employer': employer,
        'job_title': job_title,
        'place': place,
        'closing_date': closing_date,
        'publishing_date': publishing_date,
        'job_description': job_description
    }

# Open the initial website
base_url = "https://www.evi.gv.at/as?page="
category = "Internationale+Jobs"
total_jobs_to_scrape = 500
job_urls = []

driver.get(base_url + "1&filter=Kategorien%3A" + category)
time.sleep(3)

# Initialize the current job number
current_job_number = 1

# Main scraping loop
while len(job_urls) < total_jobs_to_scrape:
    page_source = driver.page_source
    soup = BeautifulSoup(page_source, 'html.parser')

    job_cards = soup.find_all('div', class_='Card-sc-ajplbe-0')
    for card in job_cards:
        if len(job_urls) < total_jobs_to_scrape:
            job_url = "https://www.evi.gv.at" + card.find('a')['href']
            job_urls.append(job_url)
        else:
            break

    try:
        next_page_button = driver.find_element(By.XPATH, '//button[@aria-label="Eine
```

```
Seite vor"]')
        driver.execute_script("arguments[0].click();", next_page_button)
        time.sleep(3)
    except:
        break  # No more pages

# Extract job details and compile data
job_data = []
for url in job_urls:
    try:
        job_details = scrape_job_details(url)
        job_data.append(job_details)
        print(f"Harvested Job {current_job_number}: {job_details['publishing_date']},
{job_details['job_title']}")
        current_job_number += 1
        time.sleep(2)
    except WebDriverException as e:
        print("Error encountered. Attempting to continue...")
        driver.quit()
        driver = webdriver.Chrome()
        driver.get(url)
        time.sleep(3)

    # Save to CSV every 50 jobs
    if current_job_number % 50 == 0 or current_job_number == total_jobs_to_scrape:
        df = pd.DataFrame(job_data)
        df.to_csv(f'job_listings_{current_job_number // 50}.csv', index=False)
        job_data = []  # Reset for the next batch

# Close the WebDriver
driver.quit()
```

Code 1: WebCrawler

III. Data Wrangling

(1) Data Cleaning

During the data wrangling step, great care was taken to prepare the dataset for NLP and ML analysis. Initially, the corpus of the job postings were pre-processed to guarantee linguistic consistency. Date fields were converted into datetime format, `place` was split into no features `city` and `country`, and ChatGPT 3.5 was employed to generate a dictionary assigning each country to a region in accordance with the United Nations geoscheme ('United Nations Geoscheme', 2023). Pre-processing procedures comprised the removal of duplicates and the cleaning of job descriptions with regular expressions `re` to eliminate

superfluous non-alphanumeric characters. The dataset contained 607 documents at this point.

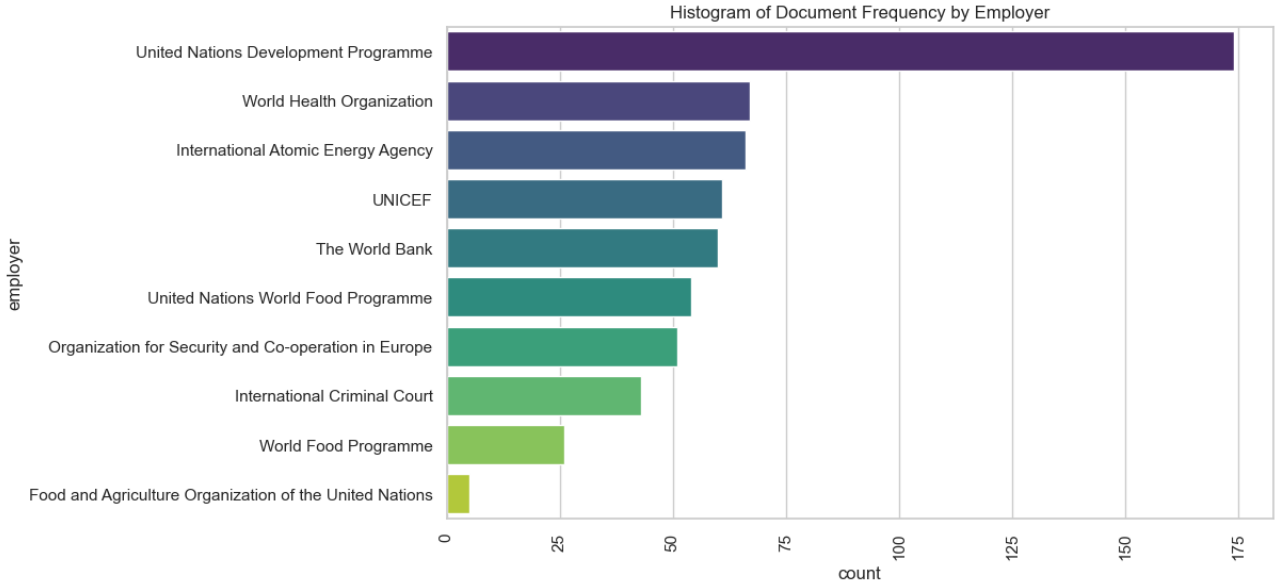


Figure 1: Histogram of Document Frequency by Employer

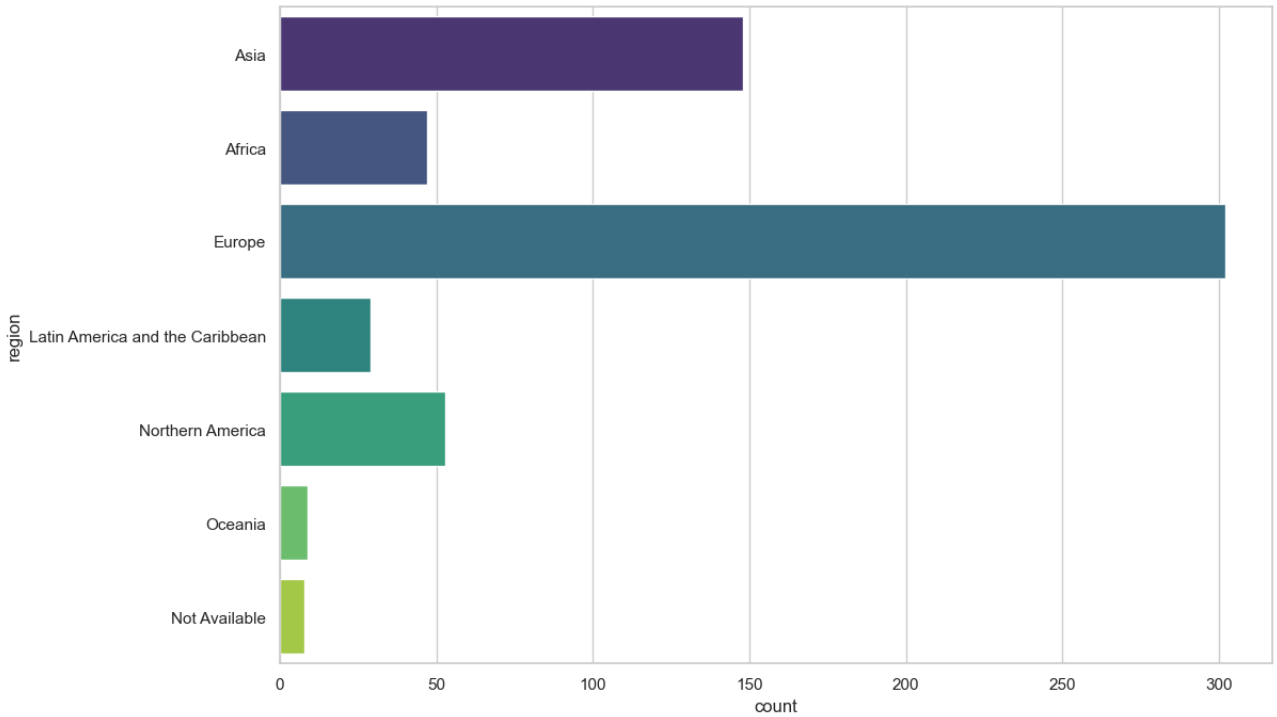


Figure 2: Histogram of Document Frequency by Region

(2) Data Normalization

Case normalisation, tokenization, stop word removal, and lemmatization were used to standardise the text data, that is `job_titles` and `job_descriptions` (Gokce, 2020; Kapadia, 2019):

- the input text was tokenized using the more advanced `spaCy`'s tokenizer with the English language model, providing better handling of complex language structures;

- stopwords were removed using `NLTK` with providing a customization option to remove additional stopwords manually (Schofield et al., 2017; *Stopwords in Technical Language Processing - PMC*, n.d.);
- lemmatization, which maintains the contextually appropriate form of a word, was applied with `NLTK`'s `WordNetLemmatizer`

To guarantee a nuanced analysis, stemming was deemed too aggressive for the task and hence omitted.

```
[('officer', 125), ('assistant', 103), ('associate', 85), ('specialist', 75),
('programme', 46), ('senior', 46), ('project', 44), ('consultant', 40),
('analyst', 39), ('communication', 34)]
--------------------------------------
[('year', 898), ('degree', 704), ('english', 631), ('management', 579),
('international', 499), ('working', 439), ('level', 435), ('field', 420),
('development', 412), ('fluency', 399)]
```

Output 1: 10 most frequent words of job titles and job descriptions

Figure 3: Word Cloud of Job Titles

Figure 4: World Cloud for Job Descriptions

After data normalization the average number of words in the `job_titles` amounted to 3.32 (i.e. 3). The average number of words in `job_descriptions` amounted to 64.99 (i.e 65).

(3) Sampling

The distribution of documents across employers (Figure 1) was given special consideration, which led to the decision to use a stratified hold-out strategy for model evaluation. This guaranteed that even employers with fewer listings were proportionally represented in the training and testing sets, minimising potential biases owing to uneven document distribution. The training set size of 485 and the test set size of 122 was created (80:20 split).

(4) Preliminary EDA

Preliminary exploratory data analysis (EDA) provided visualizations like word clouds (Figure 5) and frequency plots, which revealed the most common terms in job descriptions, and helped in understanding the distribution of job listings across `employers`. Beyond a glimpse into the data, word clouds do not provide much of an insight into potential latent patterns.

Figure 5: Word Clouds by Employer

IV. Machine Learning

(1) NLP Pipeline

Latent Dirichlet allocation (LDA) is a generative topic model to find latent topics in a text corpus. (Chen & Wang, n.d.)

LDA is a generative probabilistic modelling strategy to reveal latent semantic structures in a set of textual sources. The basic idea is that each document contains a mixture of latent topics, each of which is distinguished by a distribution over the words (unique words in the collection of documents), and the relative importance of the topics varies from document to document (Bastani et al., 2019). We used LDA's probabilistic inference approach to find word clusters and, thus, themes from a collection of words in documents (job listings), as well as the most relevant common topics throughout a corpus of documents. The technique has a significant benefit over other approaches to surveying a literature in that no prior knowledge of themes is required (Aziz et al., 2022).

During the machine learning phase, an NLP pipeline (Albanese, 2022; Kapadia, 2022; Mahmood, 2023; Tran (Talia), 2023) that took in preprocessing results (→ Data Wrangling) was built:

i. Step 1

Out-of-the-box LDA using the `sklearn's CountVectorizer` to create the document-term matrix (DTM) with the parameters (Figure 7) set:

- `max_df`: used to remove terms that appear too frequently and are therefore less informative. (e.g., 0.95, meaning terms that appear in more than 95% of the documents will be ignored).
- `min_df`: removes terms that appear too infrequently to be meaningful. Like max_df, it can be set as an absolute count or a proportion of documents.
- `stop_words`: 'english', a standard package.

These parameters will be subject to tuning at a later step.

ii. Step 2

Preliminary topic exploration with the `topicwizard` package, allowing for interactive topic exploration. 4 to 8 topics appeared to be a reasonable choice in the exploration, graphically the do not overlap and thus seem to be distinct.

```
Pipeline
Pipeline(steps=[('vectorizer',
                 CountVectorizer(max_df=0.95, min_df=2, stop_words='english')),
                ('topic_model',
                 LatentDirichletAllocation(n_components=7, random_state=0))])
    CountVectorizer
    CountVectorizer(max_df=0.95, min_df=2, stop_words='english')
    LatentDirichletAllocation
    LatentDirichletAllocation(n_components=7, random_state=0)
```

Figure 6: topicwizard model paramaters

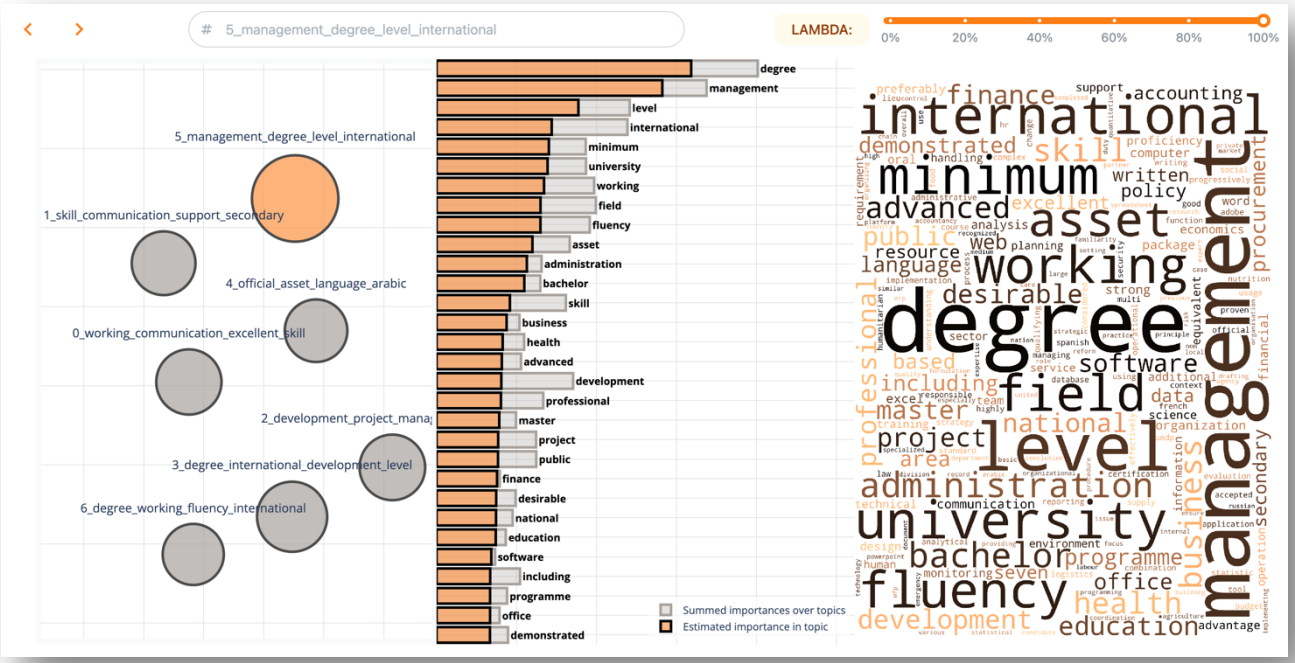


Figure 7: topicwizard "Topics"

Figure 8: topicwizard "Words"

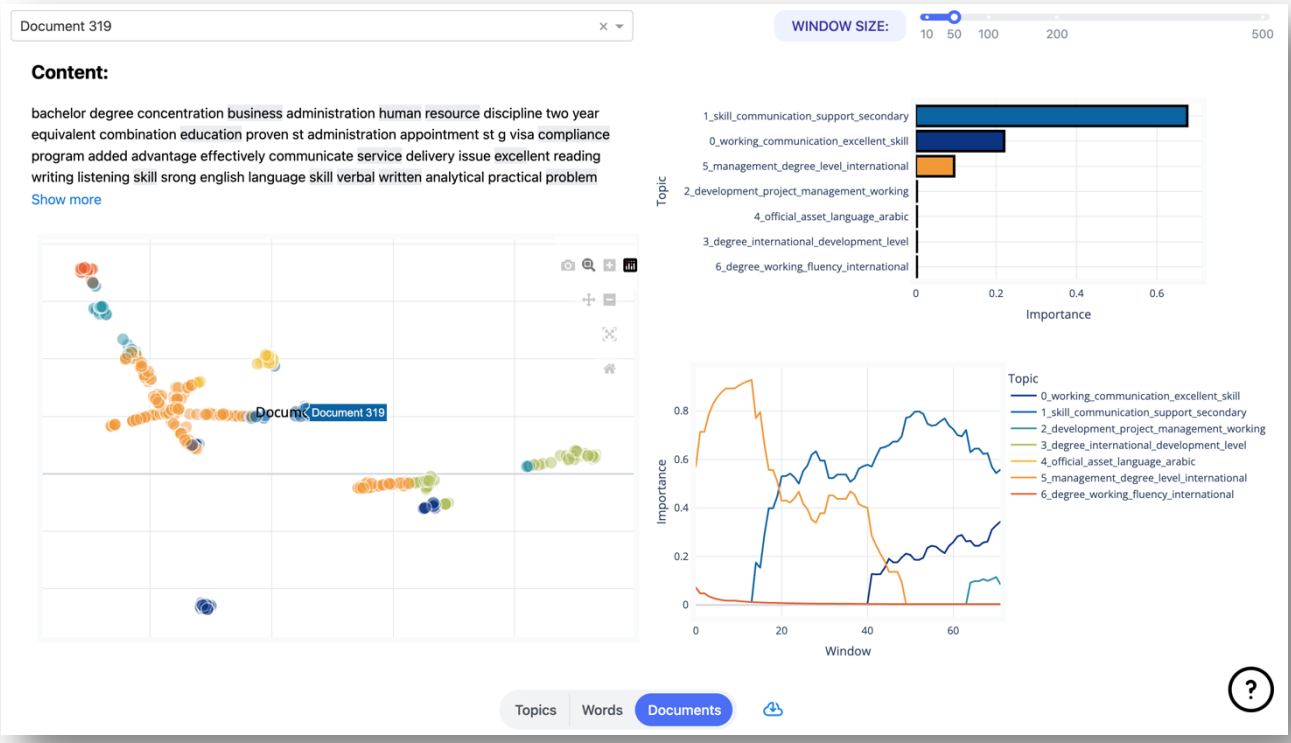


Figure 9: topicwizard "Documents"

iii. Step 3: Topic Distribution by Employer

The first topic distribution suggests that the model can be further improved. Topic 5 appears to be strongly present across employers.

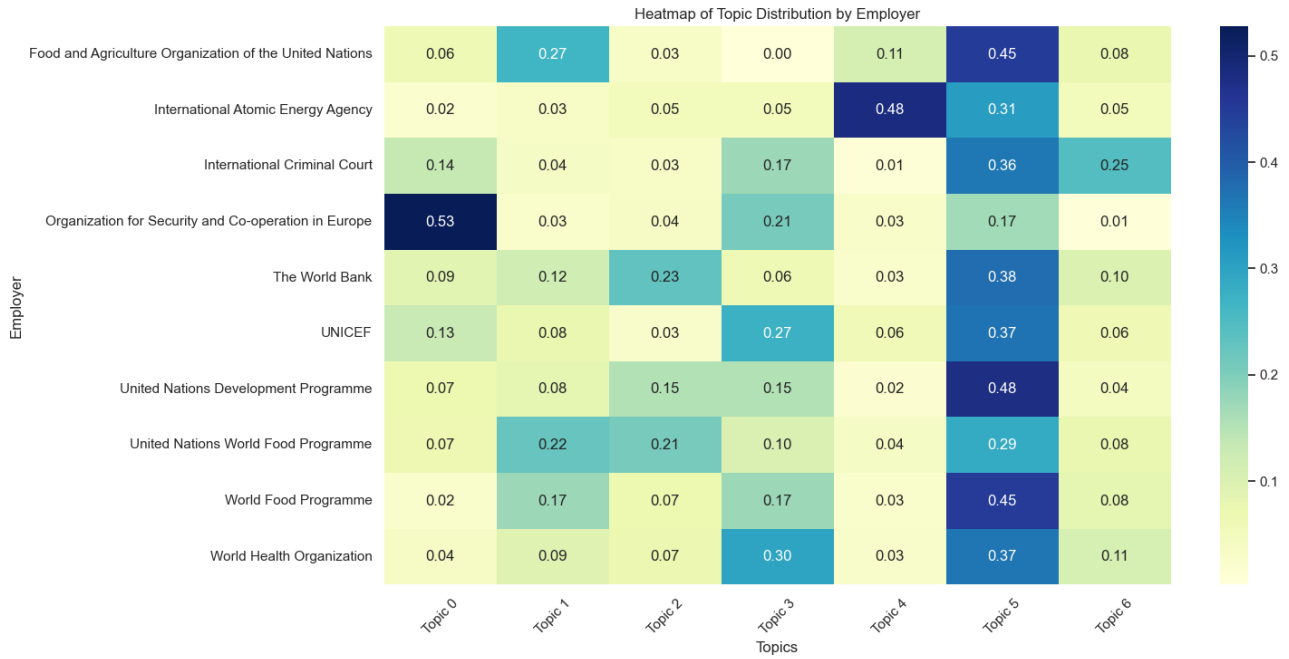
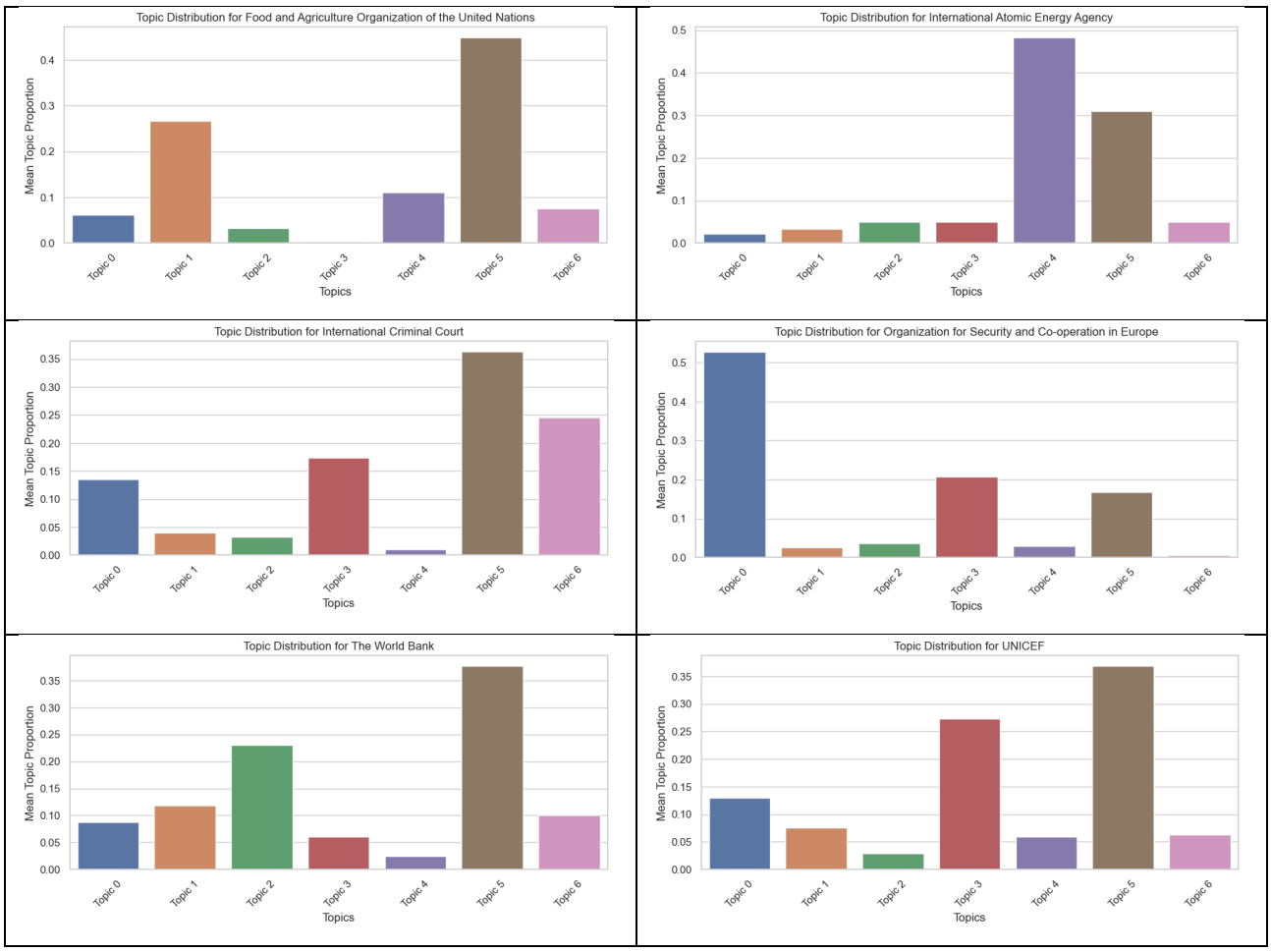


Figure 10: Heatmap of Topic Distribution by Employer



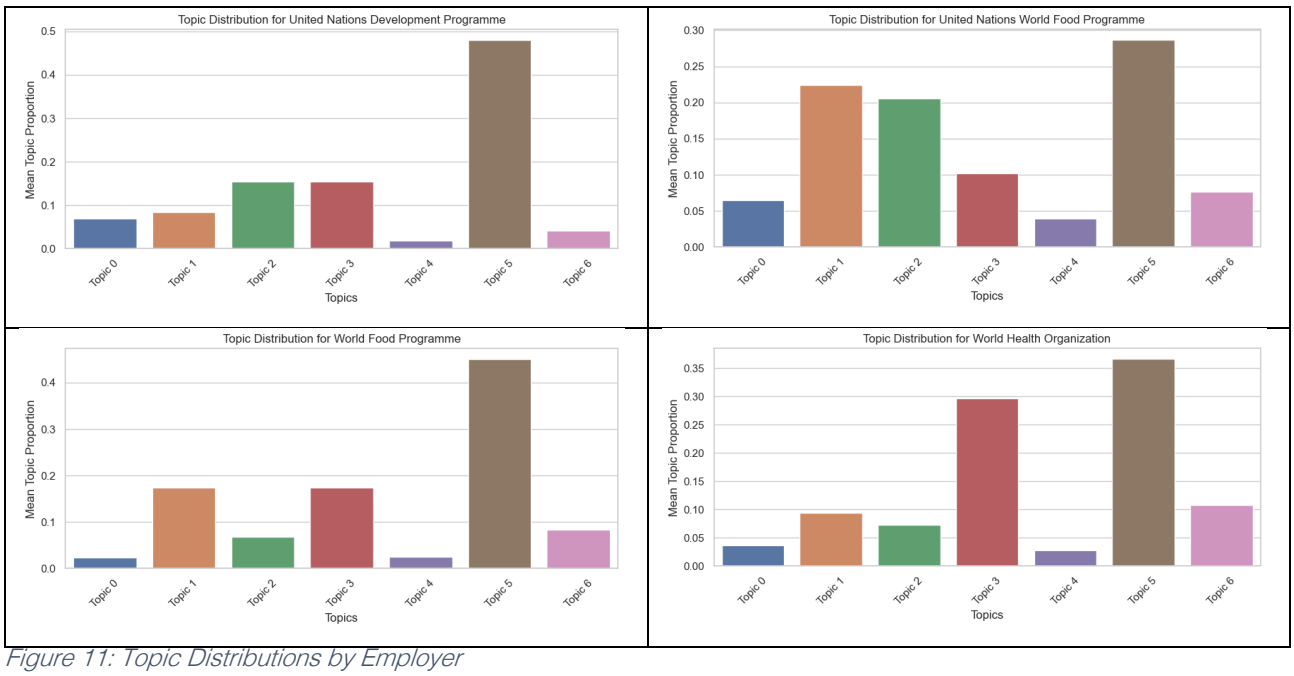


Figure 11: Topic Distributions by Employer

iv. Step 4: Grid Search: Optimal number of topics

We conduct a grid search to find an optimized number of topics. The coherence score and perplexity[1] supported us in the decision making. We decided to go with 5 topics, striking a sensible balance between both scores.

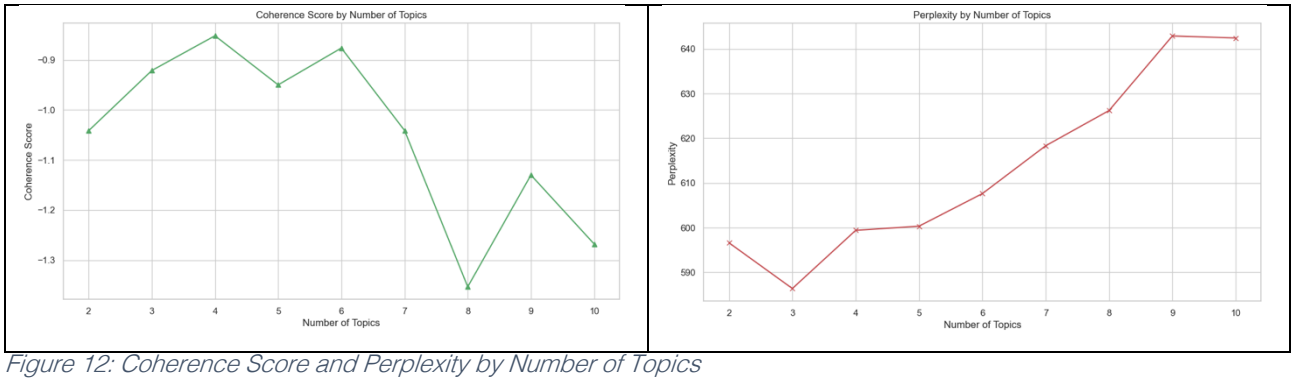


Figure 12: Coherence Score and Perplexity by Number of Topics

v. Step 4: Grid Search: Extended hyperparameter tuning

Further model parameters (`max_df`, `min_df`, `topics`) were determined in a grid search.

```
#%%
# Extracting the lemmatized job descriptions
lemmatized_descriptions = df['job_description_lemmatized']

# Define the ranges for max_df, min_df, and number of topics
max_df_values = [0.60, 0.95]
min_df_values = [2, 5]
topics_range = range(2, 11)  # From 2 to 10 topics

# Store all results
all_results = []
```

[1] For details of the scores refer to (ix) Performance

```
for max_df_val in max_df_values:
    for min_df_val in min_df_values:
        # Vectorize the text data with current max_df and min_df values
        vectorizer = CountVectorizer(max_df=max_df_val, min_df=min_df_val,
stop_words='english')
        dtm = vectorizer.fit_transform(lemmatized_descriptions)
        terms = vectorizer.get_feature_names_out()

        # Convert sparse matrix to gensim corpus
        corpus = Sparse2Corpus(dtm, documents_columns=False)
        # Create Gensim dictionary from the terms
        id2word = Dictionary([terms])

        for num_topics in topics_range:
            # Create and fit LDA model
            lda = LatentDirichletAllocation(n_components=num_topics, learning_decay=0.7,
random_state=0)
            lda.fit(dtm)

            # Coherence Score
            lda_topics = [[terms[i] for i in topic.argsort()[:-11:-1]] for topic in
lda.components_]
            cm = CoherenceModel(topics=lda_topics, corpus=corpus, dictionary=id2word,
coherence='u_mass')
            coherence = cm.get_coherence()

            # Perplexity
            perplexity = lda.perplexity(dtm)

            # Append results
            all_results.append({
                'max_df': max_df_val,
                'min_df': min_df_val,
                'num_topics': num_topics,
                'coherence': coherence,
                'perplexity': perplexity
            })

# Convert results to DataFrame for better visualization
results_df = pd.DataFrame(all_results)
print(results_df)
```

Code 2: Grid search

Following the grid search, we normalized the coherence scores and perplexities and combined them to receive a balanced approach of the best combinations. We chose model No° 30.

```
    max_df  min_df  num_topics  coherence  perplexity
27    0.95       5           2  -0.829548  386.297120
30    0.95       5           5  -0.937791  387.929712
28    0.95       5           3  -0.962632  384.121012
```

Output 2: Top 3 models according to balanced coherence/perplexity approach

(2) LDA Implementation

With the model parameters determined, the LDA implementation ran through the following steps.

vi. Fitting the final model

The fitted model returned the following topics with respective weights[2].

```
Topic 0:
asset (128.55) language (97.09) support (66.40) management (65.65) arabic
(64.57) official (61.77) french (59.51) information (58.48) administrative
(55.32) spanish (53.47)

Topic 1:
data (106.93) working (57.62) development (43.75) analysis (39.03) international
(35.65) language (34.48) oral (29.13) written (28.46) communication (26.45)
fluent (24.98)

Topic 2:
degree (235.23) international (200.83) university (178.22) field (138.18)
communication (132.25) professional (120.00) management (97.73) relation (95.98)
working (95.02) advanced (91.86)

Topic 3:
management (118.71) degree (84.30) health (78.81) international (77.83)
administration (76.08) finance (65.75) accounting (60.20) procurement (59.72)
office (54.55) web (52.33)

Topic 4:
development (221.84) degree (190.54) management (173.89) project (147.37) master
(113.53) field (105.27) working (100.67) policy (98.44) science (97.88)
international (87.18)
```

Output 3: Topics with topic weights for the selected, final model

vii. Analyzing job titles for each topic and qualitative interpretation

To inform decision makers, interpreting topics against the backdrop of identified job roles involved correlating the significant words in each topic with the common roles derived from job titles. The aim was to align the thematic emphasis of each topic with the practical job roles in the institution:

Topic	Primary Focus and Key Attributes	Likely Roles and Responsibilities
Topic 0	Asset Management, Multilingual Support, and Administrative Efficiency	Roles include Administration Officers, Administrative Assistants, and Support Officers. Key responsibilities involve asset management, multilingual administrative support (Arabic, French, Spanish), and efficient handling of information.
Topic 1	Data Analysis and Development with Multilingual Communication Skills	Positions like AIDS Programme Assistants, Analysts, and Evaluation Analysts. Focus on data analysis, development projects, and effective oral/written communication in multiple languages.

[2] Higher weights indicate that a word is more important or more central to that topic. They are relative measures within each topic and can be influenced by the frequency of words in the documents, the number of documents, and the number of topics in the model.

Topic 2	Advanced Degrees, International Communication, and Professional Management	Access and Acquisition Officers, Gender Advisers, and Administrative Finance Associates. Emphasis on international expertise, advanced university degrees, and professional management in fields like gender studies and international relations.
Topic 3	Health and Financial Management with a Focus on Administration and Procurement	Roles in accounting, finance administration, and procurement such as Accounting Assistants and Administrative Associates. Involvement in health management, finance, and office administration.
Topic 4	Project Development and Management with a Science and Policy Orientation	Academy Managers, Advisors, and Agricultural Specialists. Central to project development and management, with expertise in fields like agriculture, policy, and science. Emphasis on master's level education and international experience.

Table 1: Matching topics with job titles

This interpretation aligns the specific skills and themes identified by the LDA model with the roles commonly associated with each topic, providing insights into the nature of job positions and the skill sets demanded in these thematic areas:

- Topic 0: Focuses on roles that require asset management and multilingual skills, particularly in Arabic, French, and Spanish. The roles are likely administrative, supporting international operations and adolescent development programs.
- Topic 1: Emphasizes data-related roles, likely involving working in international environments. The roles include data analysts and assistants in health programs (e.g., AIDS), with a focus on oral and written communication skills.
- Topic 2: Centers around roles requiring higher education degrees and professional experience in communication. These positions could be in support, acquisition, and advisory capacities, dealing with gender and international relations.
- Topic 3: Relates to roles in management, health administration, finance, and accounting. It suggests positions responsible for procurement, office administration, and finance-related tasks.
- Topic 4: Deals with development projects and policy-making, requiring advanced educational qualifications. Roles may include managers, advisors, and specialists in advocacy, marketing, and agricultural development.

To facilitate our interpretations, we utilized Figure 14, Figure 15, Figure 16 below.

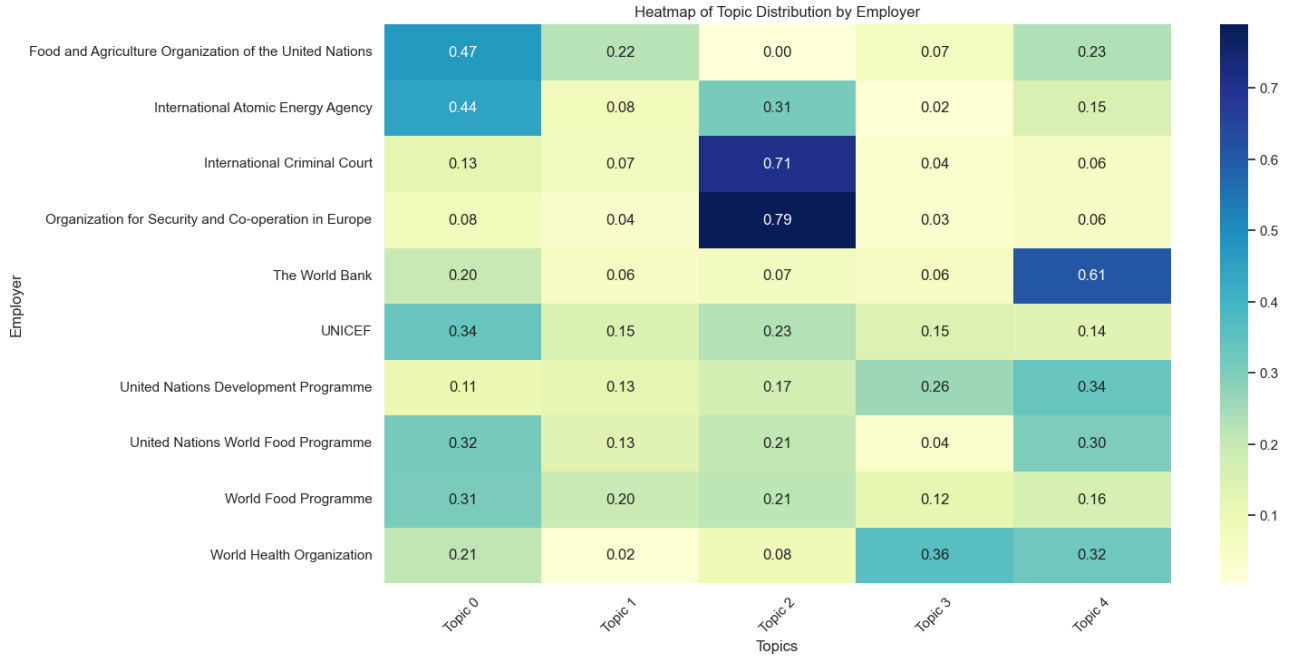
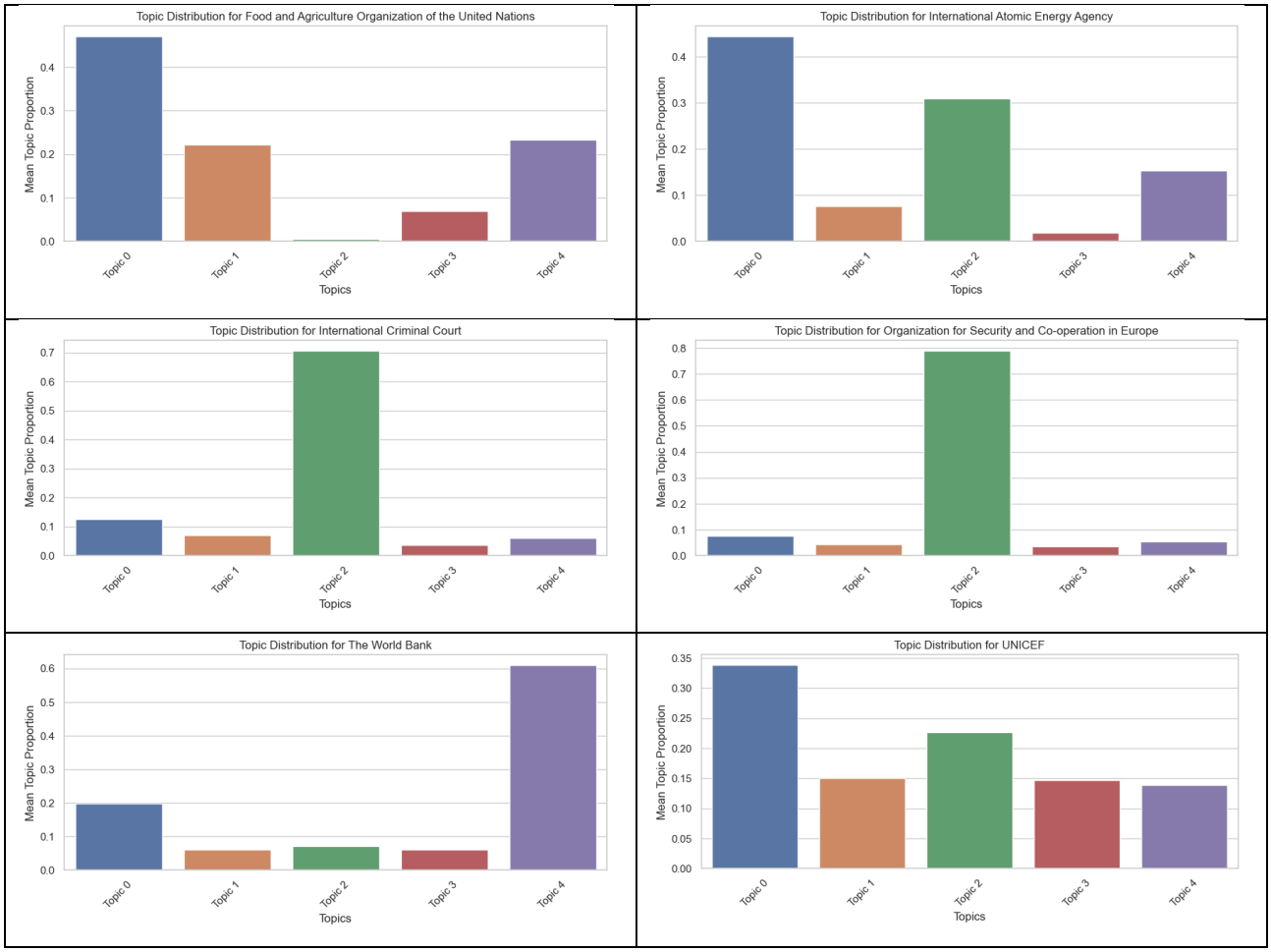


Figure 13: Heatmap of Topic Distribution by Employer (final model)



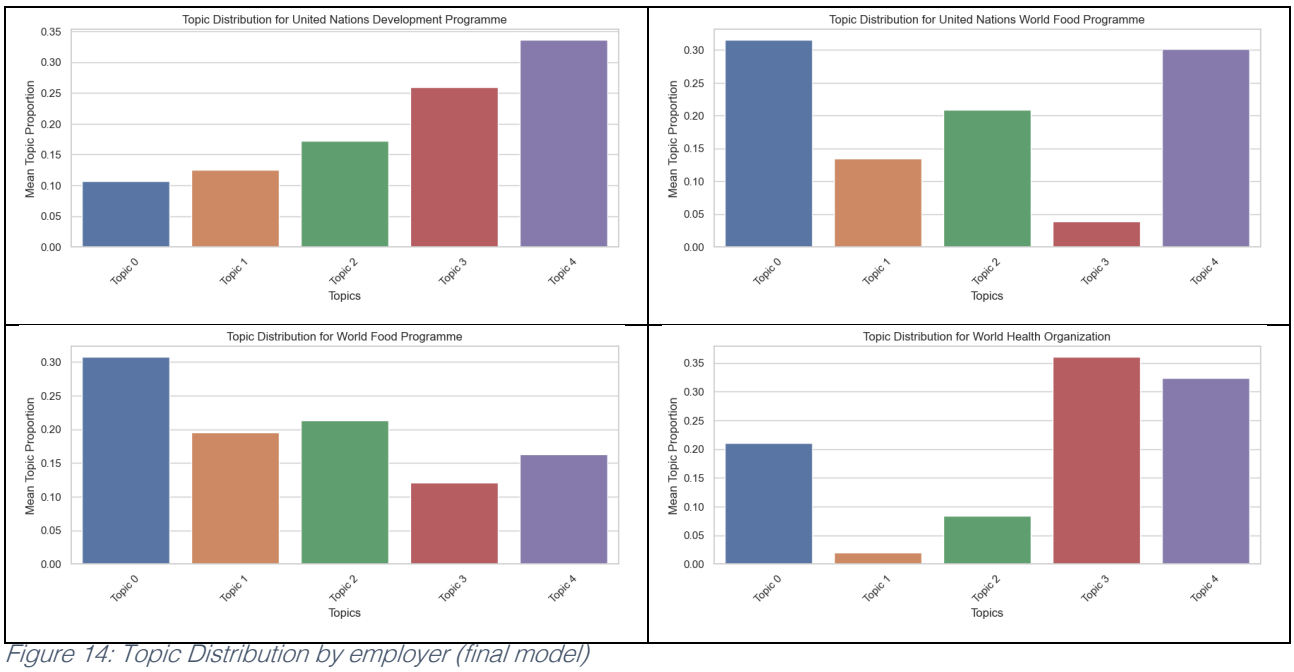


Figure 14: Topic Distribution by employer (final model)

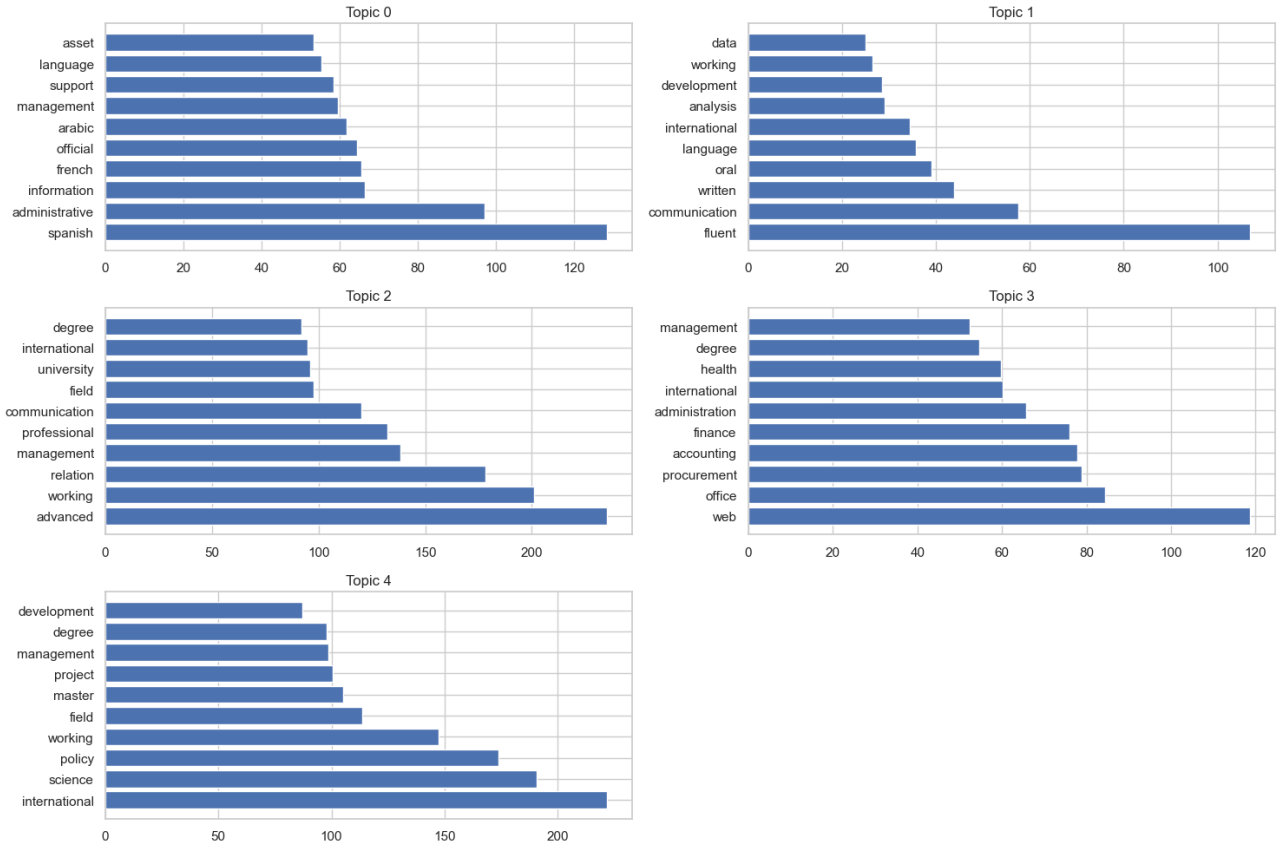


Figure 15: Top words in each topic for the final model

viii. Final interpretation by employer

The observed distribution patterns make sense within the context of each employer's domain. It's evident that different organisations have varied thematic focuses in their job advertisements, reflecting their respective goals and the types of tasks they're looking to fill. various findings give useful insights that can be used to inform recruitment strategies, training, and educational alignment for workforce growth in various areas.

Organization	Interpretation of topics and words

Food and Agriculture Organization of the United Nations (FAO)	Seems to have a strong emphasis on Topic 0, which could be related to agricultural assets management, language support, and administrative efficiency, considering the top words associated with this topic. This makes sense for an organization focused on food and agriculture, where asset management and multilingual communication are key.
International Atomic Energy Agency (IAEA)	Shows a significant emphasis on Topic 2, possibly associated with advanced degrees, international communication, and professional management. The focus on high education and international expertise aligns with the agency's need for skilled professionals in the field of atomic energy.
International Criminal Court (ICC)	Has a strong emphasis on Topic 2, has a significant need for individuals with advanced education and expertise in international relations, law, or similar fields. It aligns well with its mandate, which requires a workforce with a high level of education and the capacity to operate in a complex international legal framework, and have clear and effective dialogue both within the court and with the international community.
Organization for Security and Co-operation in Europe (OSCE)	Predominantly focuses on Topic 2, similar to the IAEA, suggesting a need for highly educated professionals with strong communication skills, which is expected for an international security-focused organization.
The World Bank	Appears to have a high emphasis on Topic 4, which might relate to development projects, advanced degrees, and policy science. This is consistent with The World Bank's role in international development and finance, where project management and policy analysis are essential.
UNICEF	Shows a diverse topic distribution with a slight emphasis on Topic 0. As a children's fund, administrative and support roles are important, but there is also a need for professionals across various fields represented by other topics.
United Nations Development Programme (UNDP)	Has a fairly even distribution with a slight emphasis on Topic 4, again aligning with their focus on development and project management.
United Nations World Food Programme (UNWFP)	Has a notable focus on Topic 0 and Topic 4, which is understandable given their work in food assistance and the need for effective asset management and development planning.
World Food Programme (WFP)	Has a similar distribution to the UNWFP, which is likely due to the overlapping nature of their work.
World Health Organization (WHO)	Shows strong emphasis on Topic 3 and Topic 4, which can be associated with health management, finance, and development. This makes sense as WHO would require professionals skilled in health finance and management, as well as those with expertise in international health development projects.

Table 2: Qualitative interpretation of topics and words in relation to employer

(3) Model Evaluation

Metrics such as coherence score and perplexity were used to test the performance of the LDA model (Lyra, 2017). While they are useful metrics it is was equally important to look at the topics qualitatively and consider other factors such as the distribution of topics across documents, and how well the topics align with our domain knowledge or objectives.

ix. Performance

The coherence score measures the degree of semantic similarity between high-scoring words in each topic (lower/less negative indicate more meaningful and interpretable topics). It is therefore a measure of interpretability. Perplexity measures how well the probability distribution predicted by the model predicts the sample and is therefore an indicator for our model's generalization capability (lower indicates better model fit).

```
Perplexity: 228.47853239931538
Perplexity (train): 397.4167380804409
Coherence Score: -15.776261765233514
Coherence Score (train): -1.382067775757747
```

Output 4: Model performance results

The coherence test score is much lower than the train score, indicating that the words within a topic are less closely related, and hence making the topic easier to interpret and understand. The coherence score of -15.78 on the test set is significantly lower than -1.38 on the training set, suggesting that the topics are much less coherent and interpretable when the model is applied to new, unseen data. This could be a sign of over-specialization on the training data or that the topics generated may not be as meaningful in the broader context of the dataset.

The perplexity of 228.48 on the test set and 397.42 on the training set implies that the model is better at predicting unseen data than fitting the training data. This suggests good generalization but might also indicate underfitting on the training data.

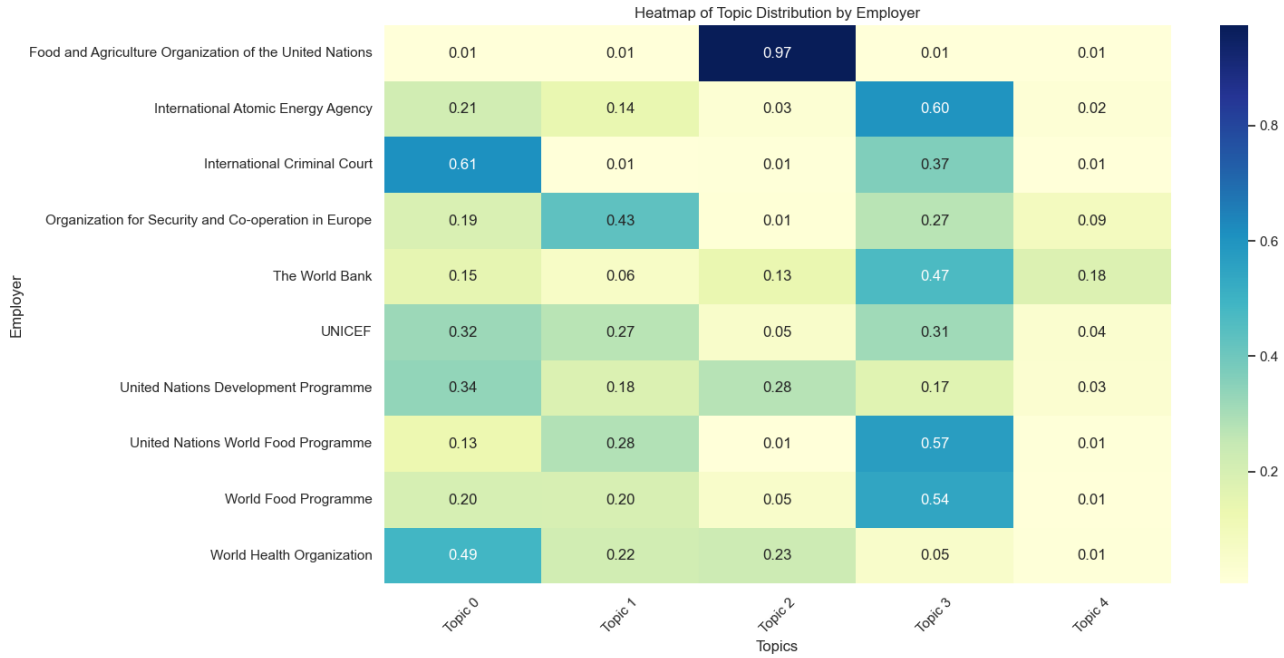


Figure 16: Heatmap of Topic Distribution by Employer for the Test Set

x. Benchmark 1: Comparing the final model to a non-negative matrix factorization (NMF) with TF-IDF

In terms of coherence scores (Coherence Score (training): 0.756, Coherence Score (test set): 0.6899), the NMF model has a significant advantage. In contrast to LDA, which had negative coherence scores in both training and testing, NMF has positive and relatively high coherence scores in both training and testing. This implies that NMF-generated topics are more interpretable and semantically consistent.

While the LDA model had good generalisation as indicated by its perplexity scores, its coherence scores dropped significantly on the test set. The NMF model, on the other hand,

maintains relatively high coherence even on unseen data, indicating good generalizability without sacrificing topic quality significantly.

```
Topic 0:
health social public communication working development international relation
degree year
Topic 1:
package finance office secondary procurement excel word education management
accounting
Topic 2:
data analysis information database collection product quality analytics using
strong
Topic 3:
asset language nuclear oral written official iaea french command russian
Topic 4:
management university degree business humanitarian training field additional
advanced human
```

Output 5: Topics of NMF model

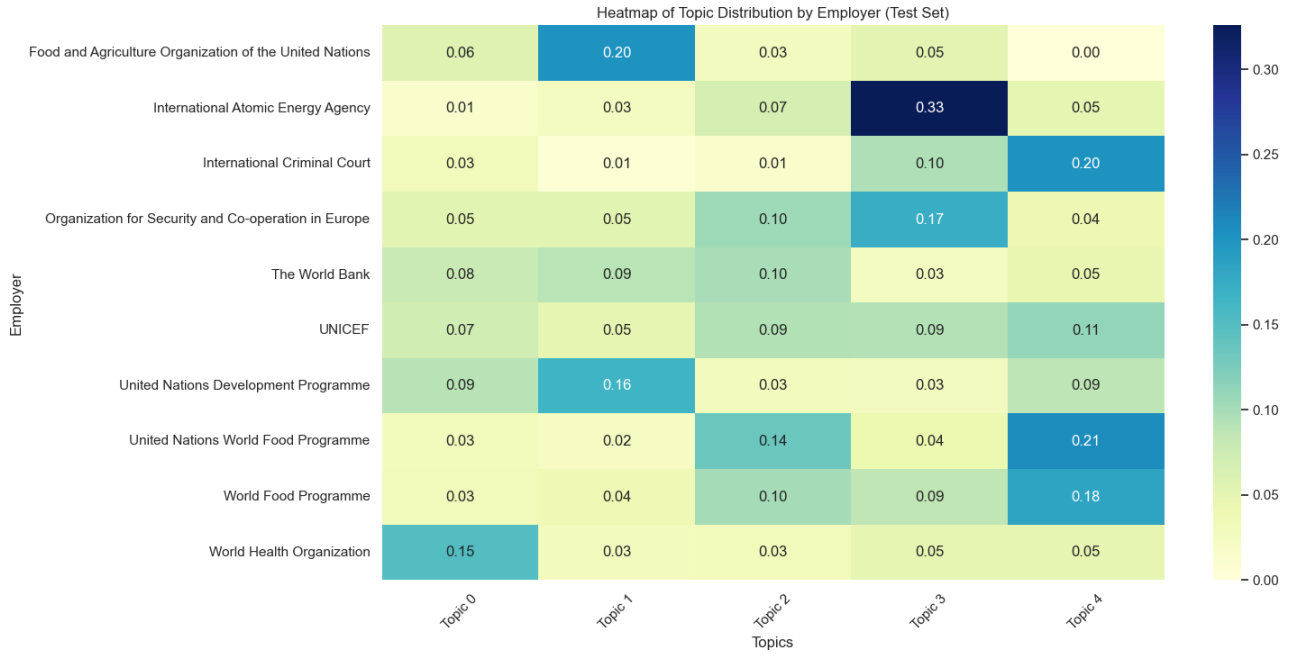


Figure 17: Heatmap of topic distribution by employer for NMF model on the test set

xi. Benchmark 2: Testing bigrams and trigrams

A derivative exploration into testing bigrams and trigrams did not yield an improvement in interpretability. As the UMAP (Uniform Manifold Approximation and Projection)[3] chart suggests, there are point clusters that indicate the presence of discrete groups of documents with comparable topics. The distinct clusters could correspond to the LDA model's identification of various prominent subjects. The distance between clusters may show how distinct or distinct various subjects are within the corpus. To reach definite conclusions, it is necessary to analyse the documents in each cluster in order to comprehend the exact issues or themes they represent (which exceeds the scope of this

[3] UMAP is a dimensionality reduction technique that is often used to visualize high-dimensional data in a lower-dimensional space, usually 2D or 3D, for easy visualization.

assessment).

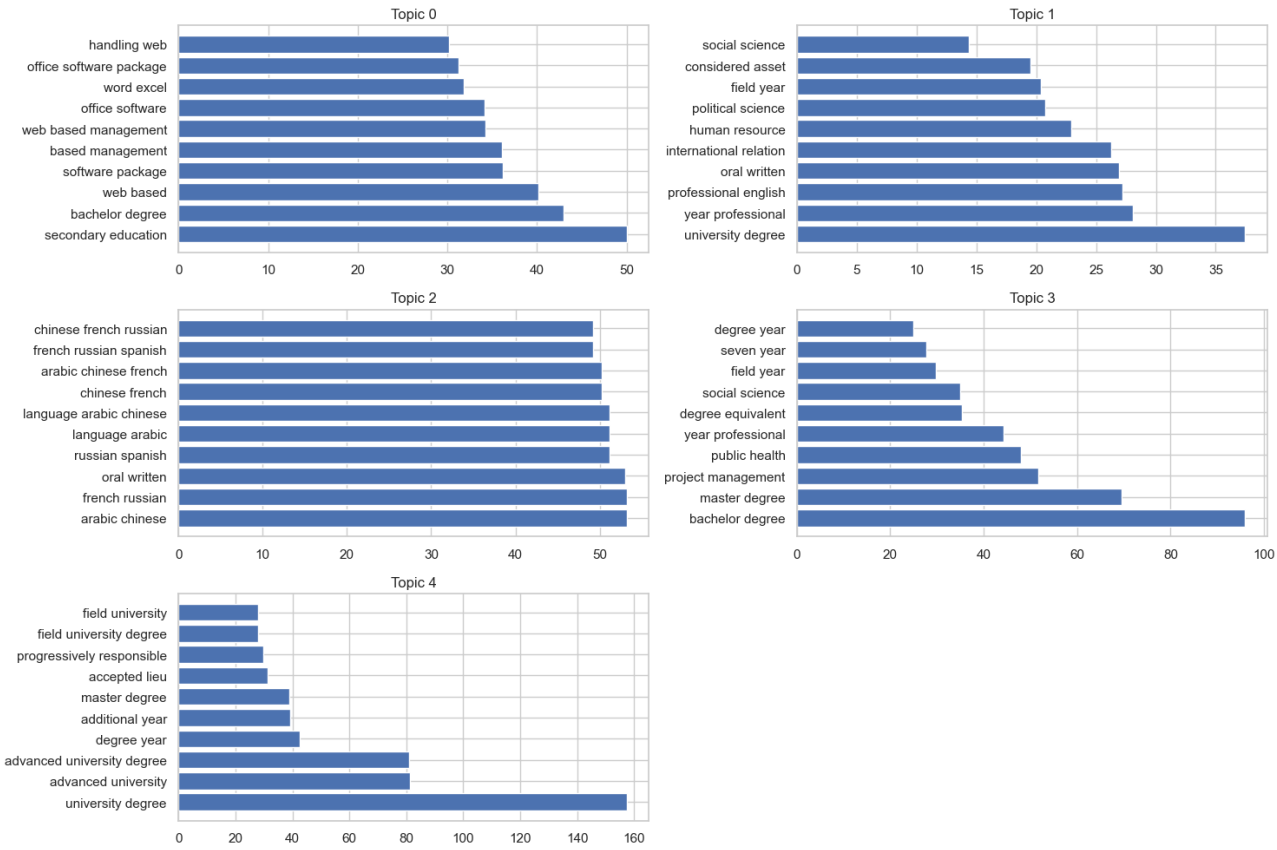


Figure 18: bigram/trigram LDA topic modeling

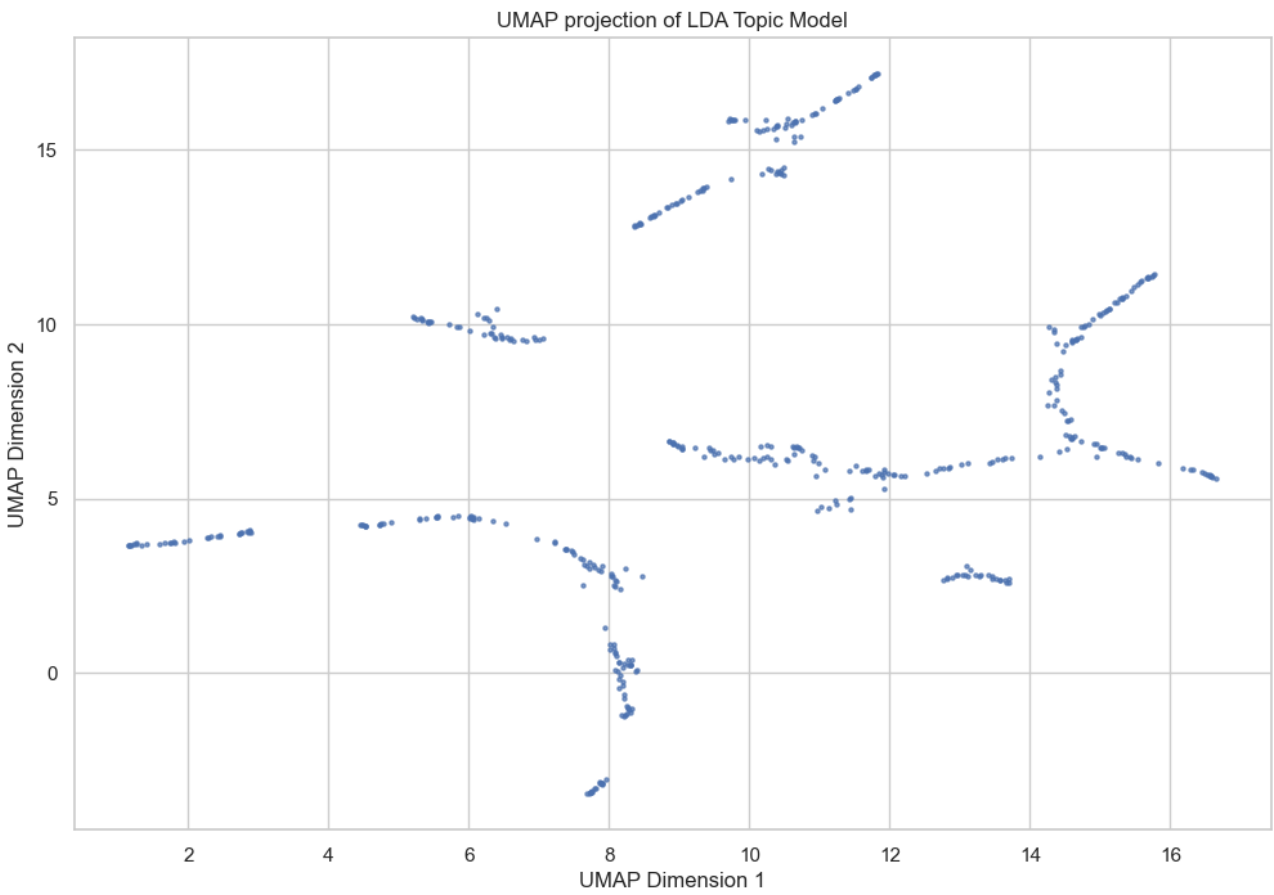


Figure 19: UMAP projection

(4) Data Limitations and Biases

The dataset's limitations and potential sample biases were acknowledged and examined in connection to their impact on the performance of the LDA model. It was emphasised that the quality of the dataset could influence the topic modelling results, particularly given the limited scope of job listings. Nonetheless, the stratified sample method assisted in reducing these biases, resulting in a more robust analysis.

(5) Conclusion

The model appears to generalise well to new data (as demonstrated by the decreased test perplexity), but the topics it creates on unseen data are substantially less cohesive. While the model is not overfitting, the subjects it highlights may be less relevant or interpretable when used outside of the training dataset. To improve the model, the parameters could be modified, the text preprocessing could be enriched, or other topic modelling techniques (such as NMF, BERTopic, Top2Vec) could be employed to better capture the thematic structures in the dataset and inform HR analytics function with valuable insights.

Based on the results, it appeared that the NMF model with TF-IDF vectorization was the better performer for the specific issue of analysing job roles and requirements. It provided clearer, more interpretable topics and maintained performance on previously unseen data. However, a final decision should take into account the specific use case as well as a qualitative assessment of the topics themselves (which exceeded the scope of this assessemtn). Even if one model produces better metrics, the other may produce topics that are more aligned and insightful with our specific analytical goals.