Capstone Project: Predicting Career Domain and Seniority from LinkedIn Profiles

Dosch Luisa, Bronner Sonia, Hüsam Laura

January 2025

Contents

1 Data Overview and Preprocessing

1.1 Label Datasets (department labels & seniority labels)

The department and seniority files were loaded into dataframes. The department labels file has 10.145 rows and the seniority labels file has 9.428 rows. Both files have 2 columns, one column corresponds to the job title, the other one to the label (department/seniority). There are 11 distinct department labels. Among them, Marketing is the most common, followed by Sales and Information Technology. Very rare are the labels Other, Purchasing, Customer Support, with Human Resources being the most rare. And there are 5 distinct seniority level labels. Among them, Senior is the most common, followed by Lead. Director, Management are less common and Junior the least. Furthermore, we found no missing values in the datasets.

1.2 CV Datasets (annotated & not annotated)

The original CV data is stored in a nested, three level hierarchical format where each CV contains a list of job entries. The datasets are a list of CVs, each CV is a list of job entries, and each job entry is a dictionary whith values. Since no job field contains nested lists or dictionaries, the hierarchy ends at the job level. Furthermore, we have 609 CVs with labels and 390 CVs without labels and there are 2.638 job titles in the annotated data and 1.886 job titles in the not annotated data.

For modeling purposes, the hierarchical CV data is transformed into a job-level table, where each row represents one job entry. For that we create the following dataframe schema where each column corresponds to a field (key) from the job dictionary:

- organization : name of the employer for a given job
- position : job title text
- startDate : start date of the job
- endDate : end date of the job
- status : indicates the status of a job
- department (annotated dataset only) : department label for a job
- seniority (annotated dataset only) : seniority level for a job

We removed the linkedin field, because it only contains a URL to a linkedin page.

Because we want to perserve the hierarchy information (CV identity and job order), we explicitly add these columns to our dataframe schema:

- cv_id : this links each job back to a CV (unique identifier for each CV)
- job_index : this preserves the job order within a CV (position of a job within a CVs job list)

This is necessary because job entries are not independent observations, they belong to the same individual and form a temporal sequence. The variables cv_id and job_index are derived from the positional indices of CVs and jobs within the nested list structure using

enumerate. All remaining columns are obtained by directly reading the corresponding fields from each job dictionary using key access.

We also inspected missing values in the datasets. Here it is important to note that some missing values are expected in our data, such as in the endDate for active jobs (status=active). Also missingness is expected for department and seniority in the not annotated data. We found 118 missing values in startDate and in 741 endDate in the annotated data and 58 in startDate and 477 in endDate in the not annotated data. We found that all missing end dates occur for jobs with status active or unkown, while all inactive jobs have an end date. Furthermore, we also found that all jobs marked with status unkown lack both a start date and an end date. This inspections showed us that incomplete temporal metadata is consistently captured via the unknown status rather than occurring randomly. This indicates that incomplete temporal metadata is consistently reflected in the job status and does not represent random missingness. Overall, any missingness in the data was expected, missing end dates correspond to ongoing (active) positions and therefore represent meaningful missingness rather than data quality issues. Jobs with unknown status lack temporal information (both start and end dates).

2 EDA

The EDA aims to understand the structure, quality, and challenges of the data.

2.1 Department Distribution

We take a look at the distribution of the department label in the datasets and understand how balanced or imbalanced the department labels are distributed. This is especially important since department is one of our target variable and its distribution helps us understand the difficulty of the prediction task.

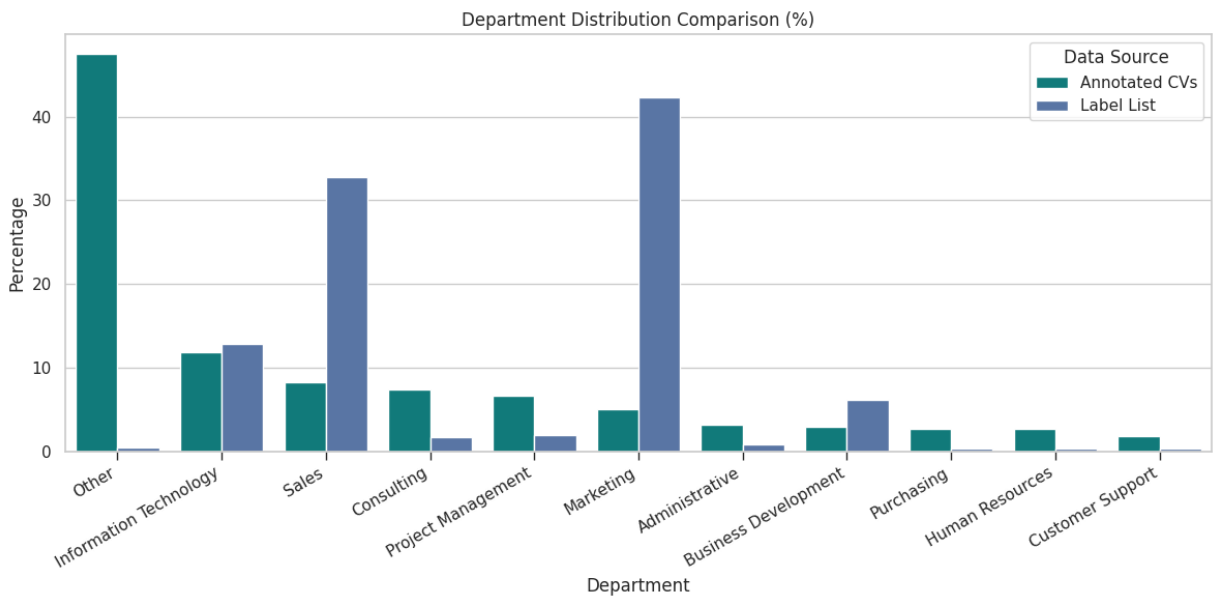

Figure 1: Department Distribution

The annotated CV data and the Label List have the same 11 departments. Departments are very imbalanced for both data. There are few departments that have a much higher percentage. However they are different for the CV data and the Label List: for the CV data the label Other dominates, while for the Label List it is Marketing. Sales and Information Technology is strong for both. The undefined label Other is much less common in the Label List than it is in the CV data.

2.1.1 How often people change Departments

We want to analyse if we can assume that it is rather unlikely that people change the department over time. We compare each job to the previous one in a CV by counting how often the department changes.

Department Change Count	CV count
0	288
1	94
2	78
3	57
4	31
5	25
6	21
7	4
8	7
9	2
10	1
24	1

Table 1: Department Change Counts and their CV Count

We can see that there is a strong department stability for many people. 288 CVs (47%) never change department and 382 CVs (63%) change at most once. But department change is not completely rare, about 37% changed department multiple times. While department changes do occur, particularly for a subset of highly mobile careers, the overall pattern indicates substantial departmental stability.

2.2 Seniority Distribution

We also take a look at the distribution of the seniority label in the datasets and understand how balanced or imbalanced the department labels are distributed. This is especially important since seniority is the other one of our target variables and its distribution helps us understand the difficulty of the prediction task.

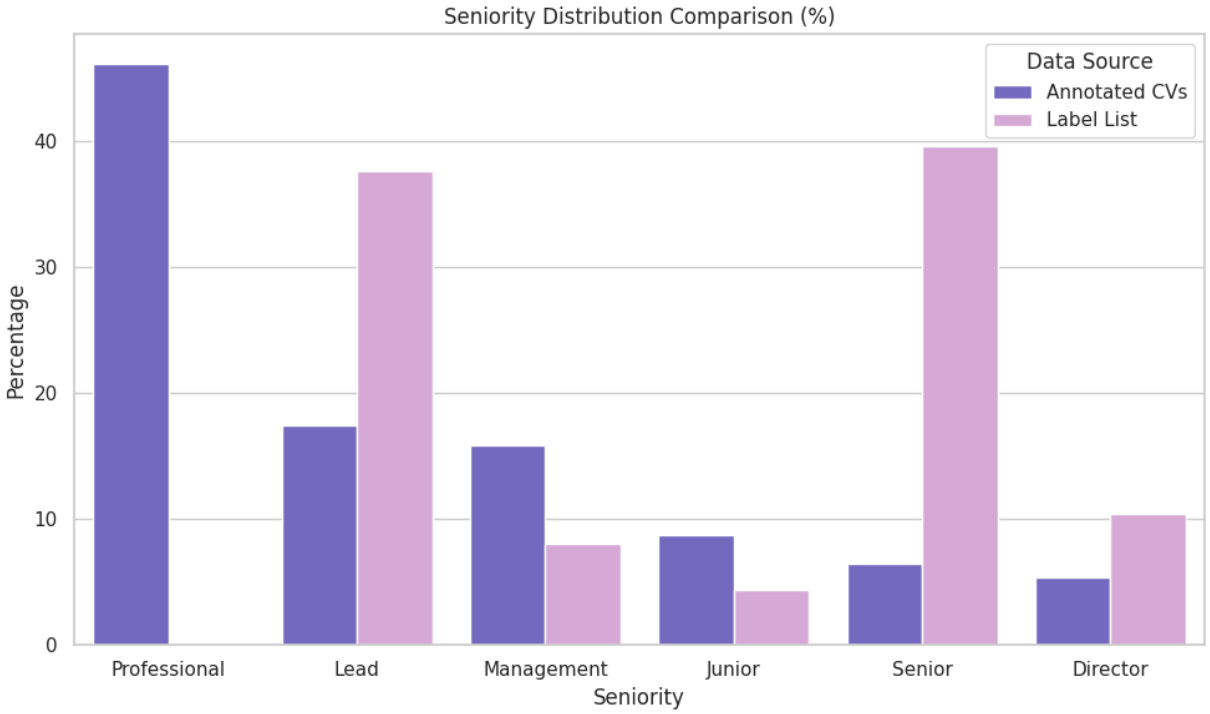

Figure 2: Seniority Distribution

The plots show a big discrepancy: we have a seniority label missing in the Seniority Label List. We have 6 seniorities in the annotated CV, but only 5 in the Label List. The label Professional is completely absend from the Label List. This is especially problematic since Professional is the most common Label in the CV data. Overall the distribution of seniority labels is more balanced compared to the distribution of department labels. Professional is the most common seniority in the annotated CVs, followed by Lead and Management. Senior is the most common seniority in the Label List, followed by Lead.

2.2.1 Comparison of current Seniority to previous Seniority

This analysis compares the seniority level of the current position to earlier roles within the same CV, highlighting whether individuals are currently in higher, similar, or lower seniority positions relative to their past experience. For that we map seniority manually to an ordinal scale (Junior = 0, Professional = 1, Senior = 2, Lead = 3, Management = 4, Director = 5) and check seniority changes between consecutive jobs in a CV.

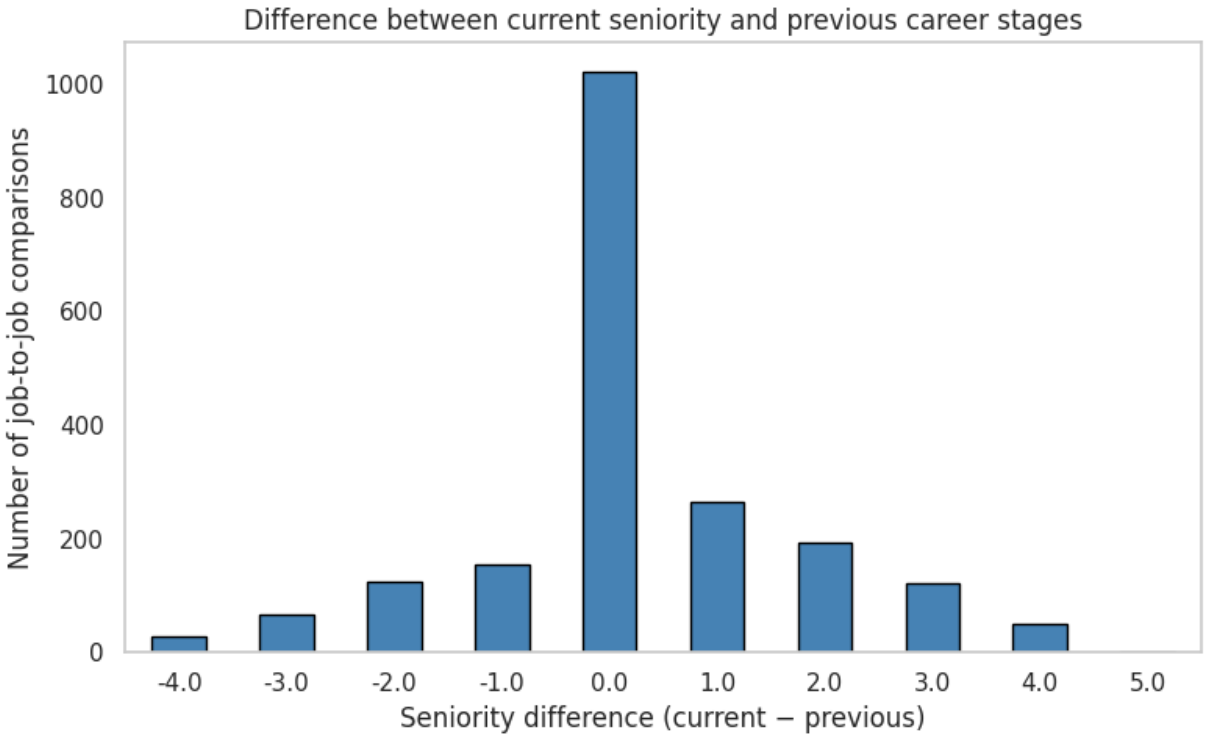

Figure 3: Seniority Changes

Job-to-Job seniority comparison	Percentage
current_higher_than_previous	31.1%
current_same_as_previous	50.37%
current_lower_than_previous	18.53%

Table 2: Job-to-Job Seniority Comparison

The majority of job-to-job comparisons show no change in seniority between the current position and earlier roles. Upward deviations are more frequent than downward deviations. This shows that seniority does not consistently increase relative to earlier roles. A large proportion of current positions remain at the same seniority level (50%).

2.2.2 How many years do people spend on each Seniority Level

We want to analyze how long (years) individuals typically remain in each seniority level. Therefore we compute the duration of job positions using their start and end dates. We restrict the analysis to inactive jobs, because we do not have a time period for active jobs. We also only use those inactive jobs with both a start and end date. Job durations are calculated in years and then aggregated by seniority level.

seniority	count	mean	std	min	25%	50%	75%	max
Director	95.0	4.563565	4.425809	0.164271	1.208761	2.997947	6.791239	22.001369
Junior	204.0	1.583605	2.342039	0.000000	0.251882	0.752909	2.336071	22.584531
Lead	289.0	4.336509	5.112513	0.082136	1.251198	2.754278	4.914442	32.418891
Management	169.0	5.462142	5.731038	0.243669	1.916496	3.915127	7.247091	40.000000
Professional	828.0	3.409090	3.935734	0.000000	1.080767	2.250513	4.162902	35.668720
Senior	112.0	4.179623	4.747329	0.251882	1.125257	2.498289	5.498289	27.496235

Figure 4: Seniority Duration

Junior positions tend to be relatively short-lived, lasting on average for one year (mean=1.58 and median=0.75 years). Professional roles represent a stable mid-career stage, with longer typical durations than junior roles but shorter durations than management or director positions. On average they last for 2-3 years (mean=3.41 and median=2.25 years). Senior and Lead roles show comparable job durations. They last on average for 2-4 years (Senior: mean=4.18 and median=2.5 years and Lead: mean=4.34 and median=2.75 years). Management positions tend to be more stable, with individuals remaining in these roles the longest. On average they last for 3-5 years (mean=5.46 and median=3.91 years). Director roles also show relatively long durations, although slightly shorter median tenure than management positions. On average they last for 3-4 years (mean=4.56 and median=2.98 years).

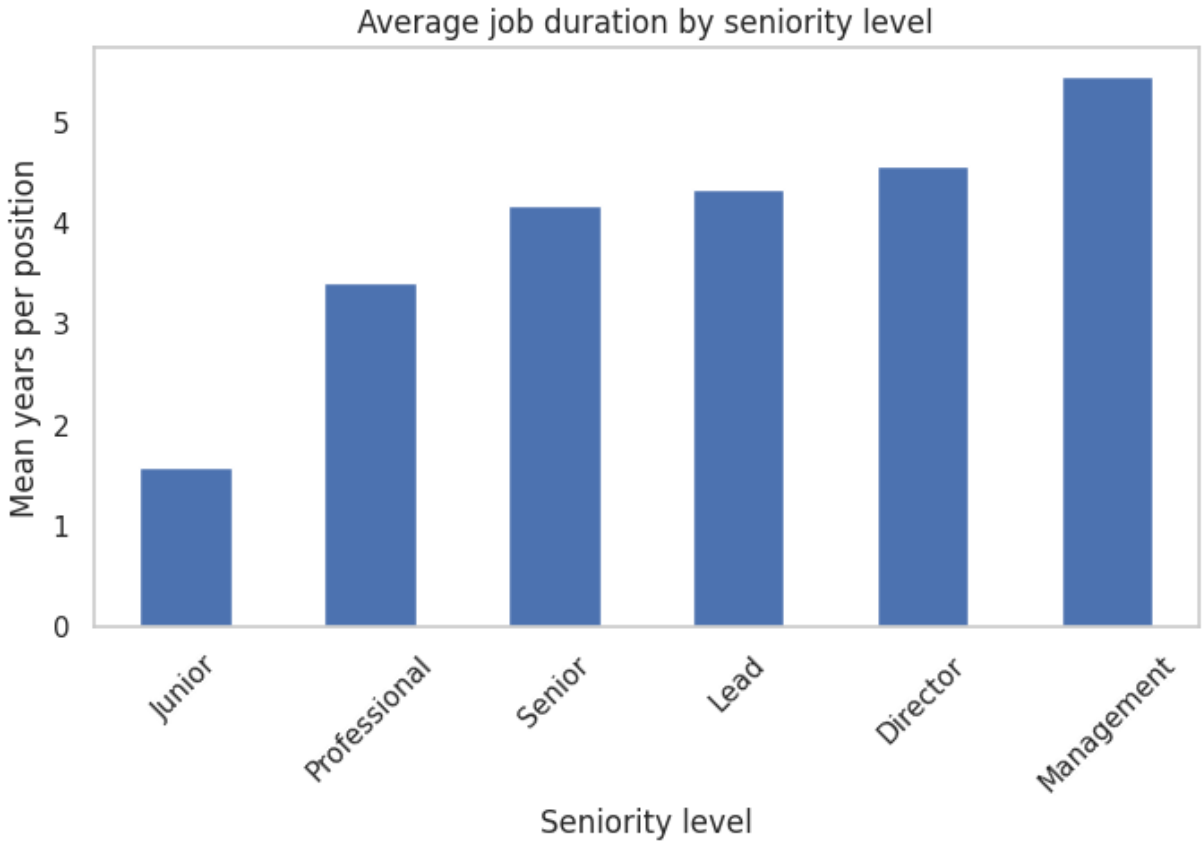

Figure 5: Average Seniority Duration

We can see that the average (mean) job duration increases with seniority level, with junior roles exhibiting the shortest average tenure and management-level roles showing the longest.

2.3 Job Status Distribution

We now take a look at the status distribution of jobs in the annotated CVs. This is important as our prediciton will be based on the characteristics of the current job, meaning where the status is active. We want to understand how many active jobs there are in the data and whether CVs can have multiple active jobs.

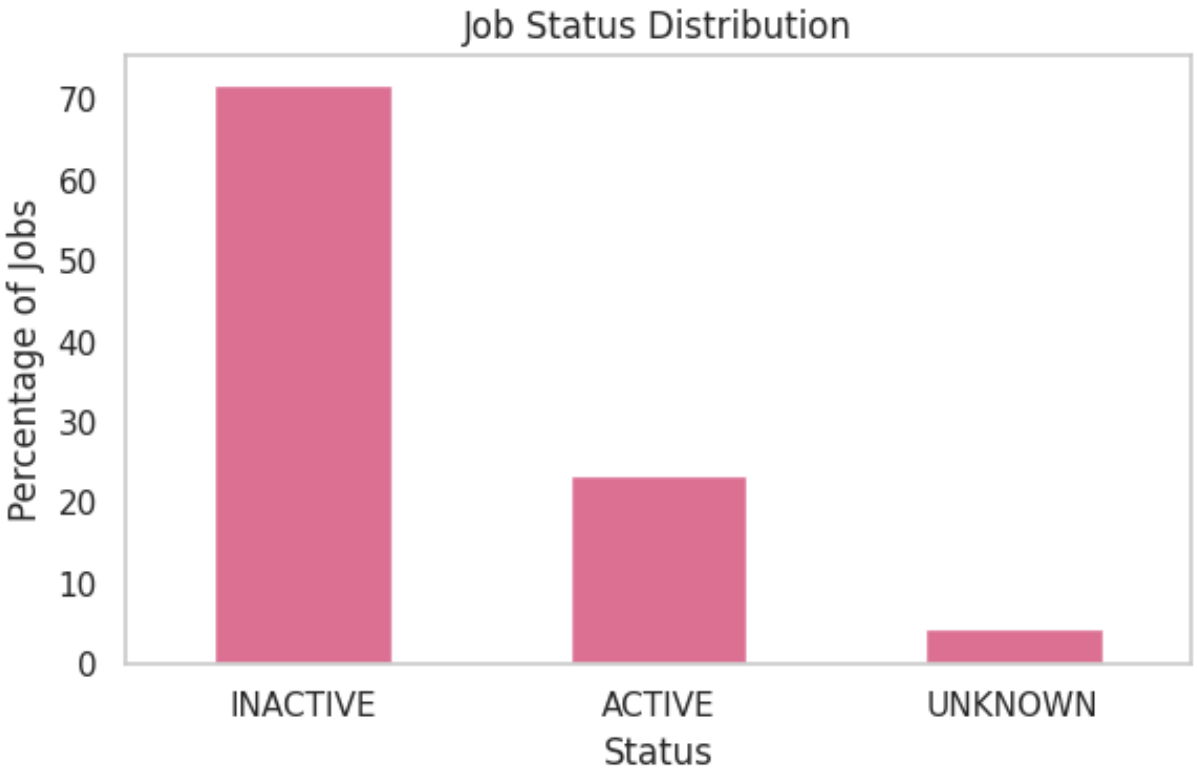

Figure 6: Job Status Distribution

We can clearly see that most jobs are inactive and historical, given by the fact that 72% of all job entries are of status inactive. This however is a logical consequence of CV data and confirms to us that the CVs contain rich career history. Approximately 24% of jobs are labeled as active, while a small fraction has a status unknown likely due to incomplete temporal metadata.

Our first intuition based on common sense and also based on the data (24% of jobs are labeled as active) is that there is 1 active job per CV. To support this intuition we take a look at the distribution of active jobs per CV.

Count of active jobs	CV count
0	131
1	380
2	78
3	10
4	3
5	4
7	1
8	1
10	1

Table 3: CV counts and their amount of active jobs

Most CVs (380) have exactly 1 active job and represents the largest group. However, we also have a high amount of CVs (131) without a current job. Some have 2 (78) or 3 (10), and there are some minor edge cases with a maximum 10 current jobs in one CV. This tells us CVs can have multiple active jobs, this however are minority cases.

2.4 Department vs Job Status

We want to understand how the departments are distributed among the different job status.

department status	Administrative	Business Development	Consulting	Customer Support	Human Resources	Information Technology	Marketing	Other	Project Management	Purchasing	Sales
ACTIVE	0.022	0.032	0.063	0.010	0.026	0.100	0.035	0.552	0.063	0.024	0.074
INACTIVE	0.032	0.031	0.079	0.022	0.026	0.127	0.057	0.438	0.071	0.030	0.089
UNKNOWN	0.076	0.000	0.051	0.008	0.034	0.085	0.017	0.661	0.017	0.008	0.042

Figure 7: Department Distribution across Job Status

Status active clearly shows a dominance of department Other. After that, IT and Sales are the strongest specific departments. This suggests that current job titles often lack strong department specific signal. Status inactive also has Other dominating but less extreme percentage wise and shows more diversity across departments. Historical jobs show broader departmental diversity than current roles. Status unknown is extremely dominated by Other and has a sparse representation elsewhere. These unknown status jobs lack sufficient structure for reliable department inference. Overall, the distribution of department labels differs by job status. Active jobs are strongly dominated by the Other category, whereas inactive jobs exhibit greater departmental diversity.

2.5 Seniority vs Job Status

We take the same look at seniority and their distribution among the different job status.

seniority status	Director	Junior	Lead	Management	Professional	Senior
ACTIVE	0.055	0.019	0.201	0.308	0.347	0.071
INACTIVE	0.055	0.114	0.167	0.104	0.495	0.064
UNKNOWN	0.017	0.017	0.153	0.237	0.542	0.034

Figure 8: Seniority Distribution across Job Status

Active jobs are strongly skewed toward mid to high seniority levels. Junior roles are rare among current jobs. Management amd Lead together alone represent 51%. Current positions tend to represent later career stages. Junior roles are much more common in inactive jobs. Earlier career stages are more prevalent in past positions.

2.6 Job Titles

Since all our approaches use the job titles as predictor, we must understand their overall structure: how long they are and how diverse / noisy they are linguistically.

2.6.1 Job Title Length Distribution

Th job title length tells us how much information is available per sample and whether titles are mostly short or descriptive. For that we measure the number of words per title.

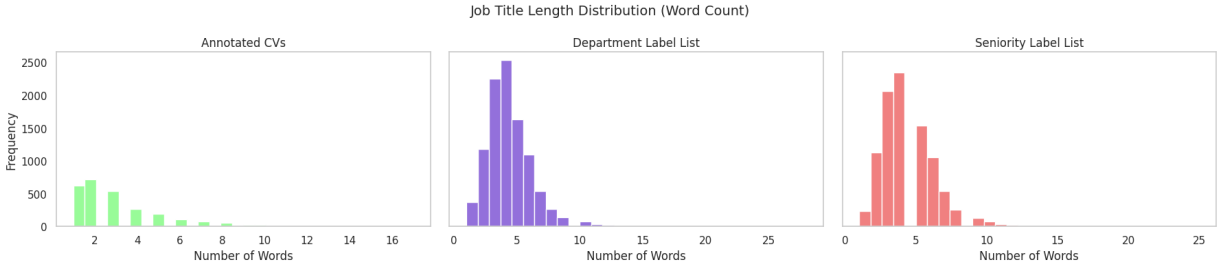

Figure 9: Job Title Length Distribution

Overall most job titles are short for all 3 datasets, ranging between 1-6 words on average per title. We can see the annotated CV data has overall shorter titles, with most of them being between 1-2 words.

2.6.2 Language and Character Diversity of Job Titles

To not overkill the analysis, we look at random samples of about 100 jobs for each dataset to get an overall view on the diversity of languages and characters in the titles. Looking at the samples, we primarily inspect the following: we analyse if non-English titles exist and we look for any special characters and whitespacing. For space reasons the full samples are only printed out in our notebook. The main insights we gained are that we have non-English titles and we have special characters (such as /, (), -, &) and whitespacing in the data for all datasets.

We also checked for the proportion of non-ASCII characters. Non-ASCII characters typically indicate: non-English titles (e.g. German, French, Spanish), localized spellings (ä, ö, ü, ß, é, ñ, etc.) or mixed-language titles.

Data	Titles with non-ASCII characters
Annotated CVs	11.1%
Department Label-List	7.1%
Seniority Label-List	9.7%

Table 4: Amount of non-ASCII characters in the job titles

For each datafile the job titles contain a small proportion of non-ASCII characters, indicating the presence of non-English or language-specific titles. This linguistic diversity introduces additional variability and noise, however the non-ascii amount is very small. Note that these are a lower bound, not an upper bound: English titles can still exist in other languages and ASCII-only does not mean “English”.

3 Preparation

3.1 Evaluation Preparation

We use the following metrics to evaluate and compare all approaches:

- Accuracy: shows the overall fraction of correctly classified samples
- macro-averaged F1-Score: is the harmonic mean of precision and recall, computed independently for each class and averaged across classes. This metric is particularly important due to the strong class imbalance in both department and seniority labels, as it ensures that performance on minority classes is weighted equally.
- MAE: measures the average absolute difference between predicted and true labels. It is only meaningful for ordinal or numerical labels (e.g. seniority levels), where it captures how far predictions deviate from the correct class on average; for nominal class labels, the metric is not applicable.

Together, these metrics allow us to assess how often each model is correct overall and how well it performs across all classes, including rare ones.

The function evaluate_predictions is created in utils for each model to use and computes the evaluation metrics. Some approaches have different metrics (numbers or strings). The function assigns None to the metric that cannot be computed for a model. This ensures a consistent output structure across different models and label types.

3.2 Results Comparison Preparation

The result_utils were created for each model to use and to add their results. In the end we can therefore get a dataframe with each model and their corresponding results to allow comparison between approaches.

The results table captures and compares across:

- Model
- Target (department vs seniority)
- Metrics (accuracy, macro F1, MAE)

Each row in that table represents one experiment.

4 Experiments

4.1 Rule Based Matching

The rule based matching is performed with the annotated CV dataset transformed to a job-level and the Seniority and Department Label Lists.

4.1.1 Preparation

For the rule-based matching, job titles and label-list entries are normalized using lowercasing and whitespace stripping before applying the substring matching. This is done to avoid any accidental mismatching due to lower/uppercasing or spacing (spaces, tabs, line breaks). We want to make sure the normalization does not alter meaning or generalize semantics, but simply removes formatting noise. Therefore, no further text processing is applied in order to keep the baseline fully interpretable. We apply the same normalization to CV titles and the label-list titles.

Furthermore, we only include CVs that contain exactly one active job (= current job) in order to ensure an unambiguous definition of the current position. CVs with zero active jobs do not provide a target position for prediction, while CVs with multiple active jobs introduce ambiguity regarding which role should be considered the primary current position. Restricting the dataset to CVs with exactly one active job therefore ensures a consistent and well-defined evaluation setup.

4.1.2 Rule-based Matching Logic

Our rule-based matching functions implement a rule-based classifier using the provided department and seniority label lists, where job titles from the LinkedIn CV data are matched against predefined keywords via substring rules. Unmatched titles to departments are assigned to the label 'Other'. For seniority we explicitly use "Professional" as default for unmatched titles, because no label such as "Other" exists in seniority. There are two reasons as to why we use "Professional" as the fallback: first, this is the most frequent seniority class and using it as the fallback label avoids artificially inflating rare classes. Second, it is present in the CV annotations but not in the predefined seniority keyword list. Whenever no seniority-specific keyword is matched, the model can default to "Professional".

4.1.3 Rule Based Matching Results

The rule-based matching baseline does not involve training and is therefore evaluated on the full dataset.

Baseline	Accuracy	Macro F1
Department prediction	60.26%	0.449
Seniority prediction	53.68%	0.426

Table 5: Rule-based matching evaluation results

Results show that for department prediction, the baseline achieves an accuracy of 60.26% and a macro-averaged F1 score of 0.449. For seniority prediction, the accuracy is 53.68% with a macro F1 score of 0.426.

The relatively higher accuracy compared to the macro F1 score in both tasks reflects the strong class imbalance present in the data. Dominant classes such as Other (department) and Professional (seniority) are frequently predicted due to the fallback behavior of the rule-based system, which inflates overall accuracy while reducing performance on minority classes.

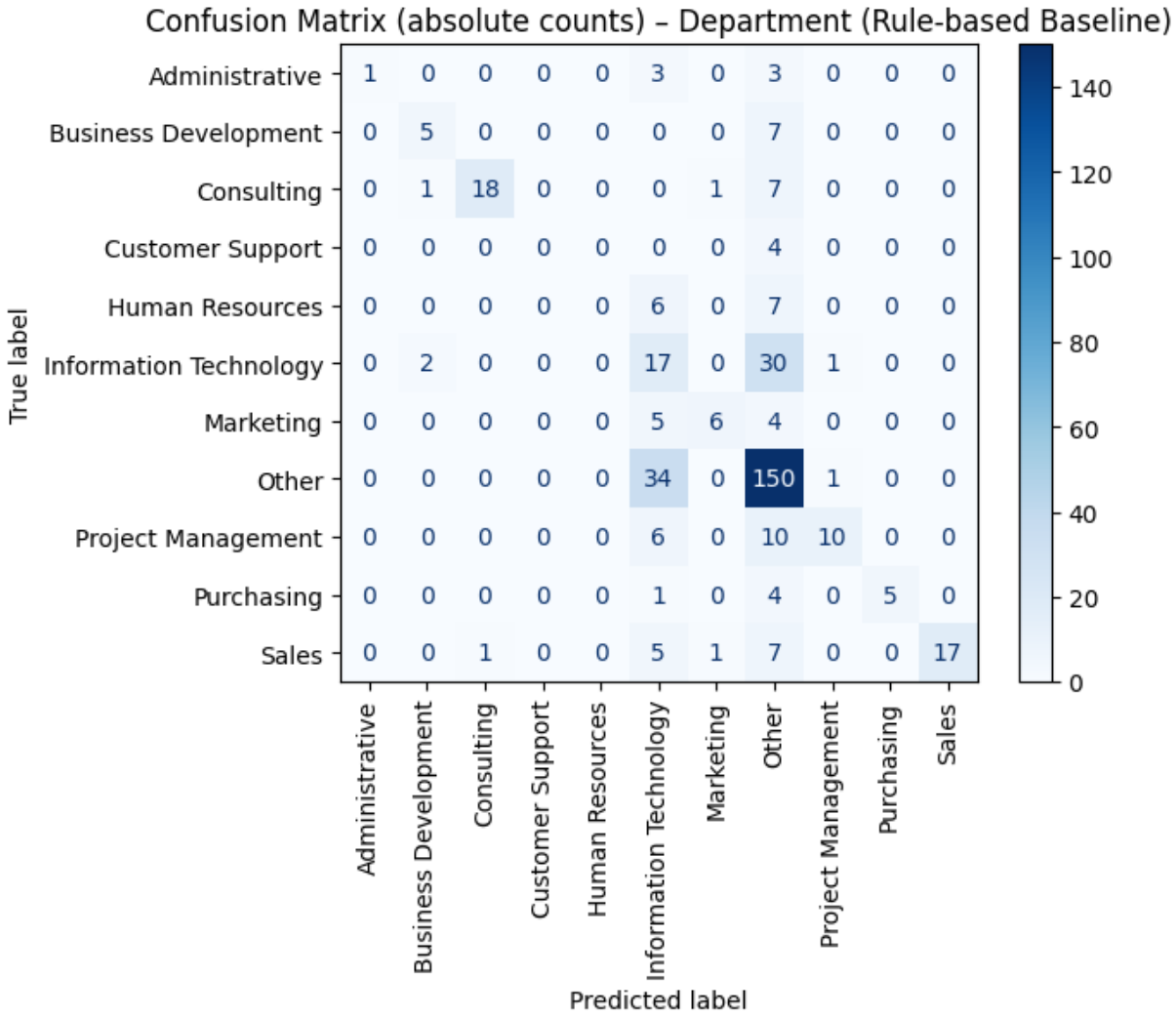

Figure 10: Confusion Matrix Department

The department confusion matrix reveals that the rule-based baseline strongly relies on explicit keywords and defaults to the “Other” category when no clear match is found. While domains such as Sales and Consulting are identified reasonably well, many roles from related or ambiguous departments are misclassified as “Other”, indicating limited contextual understanding.

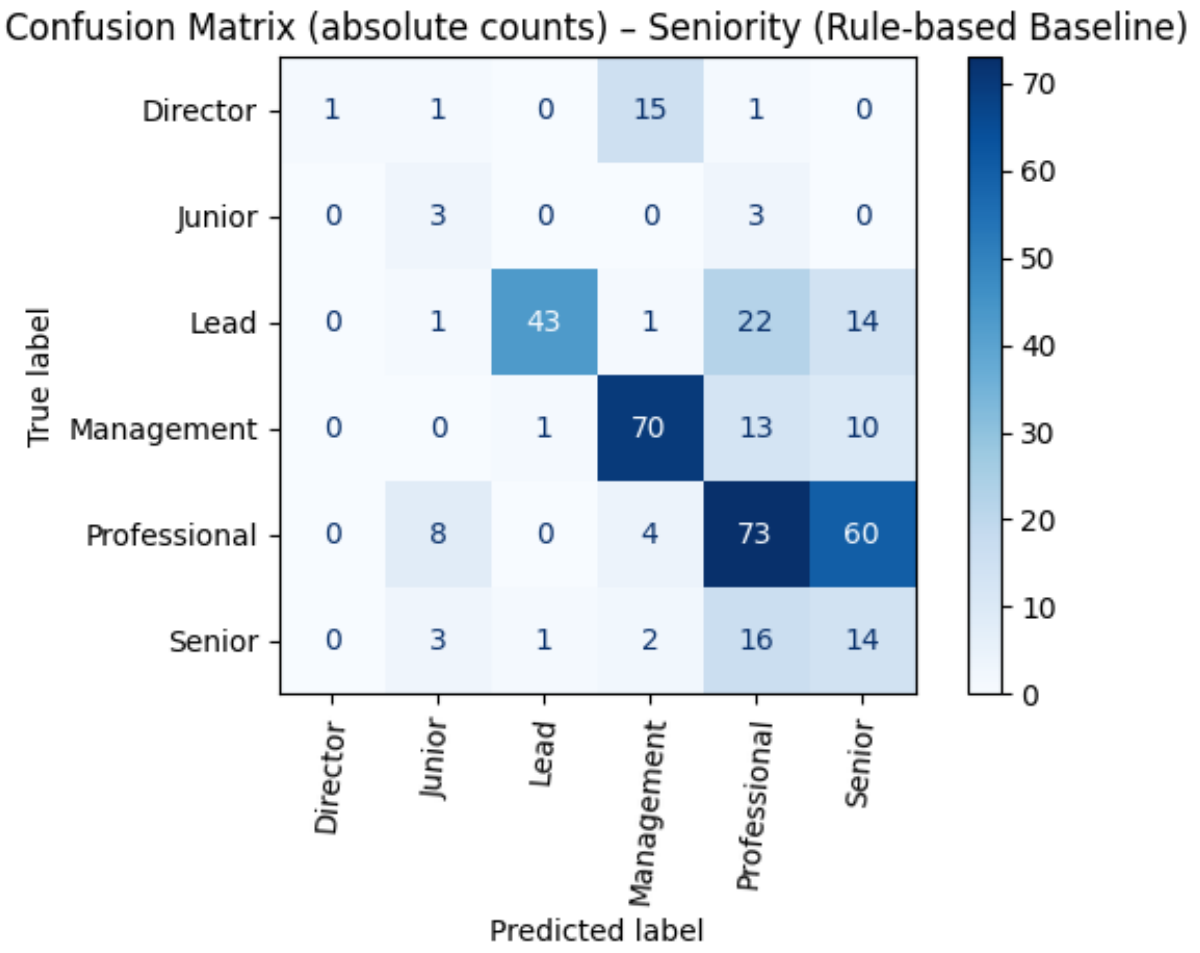

Figure 11: Confusion Matrix Seniority

The seniority confusion matrix is strongly influenced by the role of “Professional” as a fallback class in the rule-based baseline. Since many job titles do not contain explicit seniority indicators, they are defaulted to “Professional”, leading to a high concentration of predictions in this class. While seniority levels with clear keywords (e.g. Lead or Management) are recognized reasonably well, substantial confusion arises for mid-level roles. This behavior highlights the limitations of keyword-based matching and motivates the use of context-aware machine learning models.

Confusion matrix analysis shows that the baseline performs reasonably well for labels associated with explicit keywords in job titles, such as Sales, Consulting, Lead, and Management. However, substantial confusion occurs between semantically related classes and for ambiguous job titles that lack clear indicators. In these cases, predictions are often assigned to fallback categories.

Overall, the rule-based baseline provides a transparent and interpretable reference point.

While its predictive performance is limited, particularly in terms of balanced class-wise performance, it establishes a meaningful lower bound and highlights the need for more expressive, context-aware machine learning models.

4.2 Simple interpretable baseline: Bag of Words

4.3 Prompt Engineering

We use prompt engineering to benchmark how far a large language model can go on our labeling tasks without training a dedicated classifier. In addition, we use the same setup to generate synthetic labels for unlabeled job titles as extra training data for downstream experiments. The implementation is available in the GitHub repository under the folder *src/prompt_engineering/*.

4.3.1 Evaluation with Test Set

We evaluate a prompt-based approach using `gemini-2.0-flash` to predict two labels from job titles: (1) an ordinal seniority level mapped to {1.0,...,6.0} and (2) a department label from an 11-class closed set. The system prompt specifies the task and the allowed labels, includes a small set of in-prompt examples, and enforces JSON-only output. To make predictions machine-readable and consistent, each response is validated against a strict schema (both fields are restricted to predefined enums). For ambiguous titles, we apply a fixed fallback (`Other` and `2.0`). Predictions are generated row-by-row; the pipeline uses retries and persists results after each row to remain robust to intermittent API errors.

Table 6 summarizes the evaluation results. On the annotated test set, seniority prediction reaches 58.43% accuracy (macro F1 = 0.54, weighted F1 = 0.60). The per-class report shows strong asymmetries: *Junior* has very low precision (0.14) and high recall (0.83), indicating substantial overprediction of this class. *Professional* and *Lead* show the inverse pattern (precision 0.82 / 0.88; recall 0.47 / 0.41), meaning these labels are correct when predicted but are assigned too rarely. *Senior* and *Management* are comparatively stable (F1 $\approx$ 0.66 each). For *Director*, recall is 1.00 but precision is 0.36, i.e., all true Director cases are retrieved, but many non-Director titles are also mapped to this top level.

Department prediction performs better in prompt engineering, achieving 79.61% accuracy (macro F1 = 0.73, weighted F1 = 0.80). The strongest categories in the class-wise report are Sales (F1 0.87), Purchasing (0.83), Information Technology (0.82), Human Resources (0.80), and Customer Support (0.91). The weakest categories are Business Development (F1 0.36) and Administrative (F1 0.55).

Task	Accuracy	Macro F1	Weighted F1
Seniority prediction	58.43%	0.540	0.601
Department prediction	79.61%	0.734	0.804

Table 6: Prompt-engineering evaluation results on the annotated test set.

4.3.2 Prompt Engineering for Synthetic Data

<We additionally apply the same prompting setup to unlabeled job titles to generate synthetic labels as extra training data. We use prompt-based labeling instead of a purely rule-based approach because it achieved higher department accuracy on the annotated dataset and because it can produce labels that are missing (or strongly underrepresented) in the supervised training data (e.g., *Professional* for seniority). The resulting file (data/results/gemini_synthetic.csv) is concatenated with the supervised training split and used for fine-tuning transformer classifier/regressor models.

Figures 12 and 13 illustrate why this augmentation is relevant: the label distributions differ substantially between the training data and the CV (out-of-production) dataset.

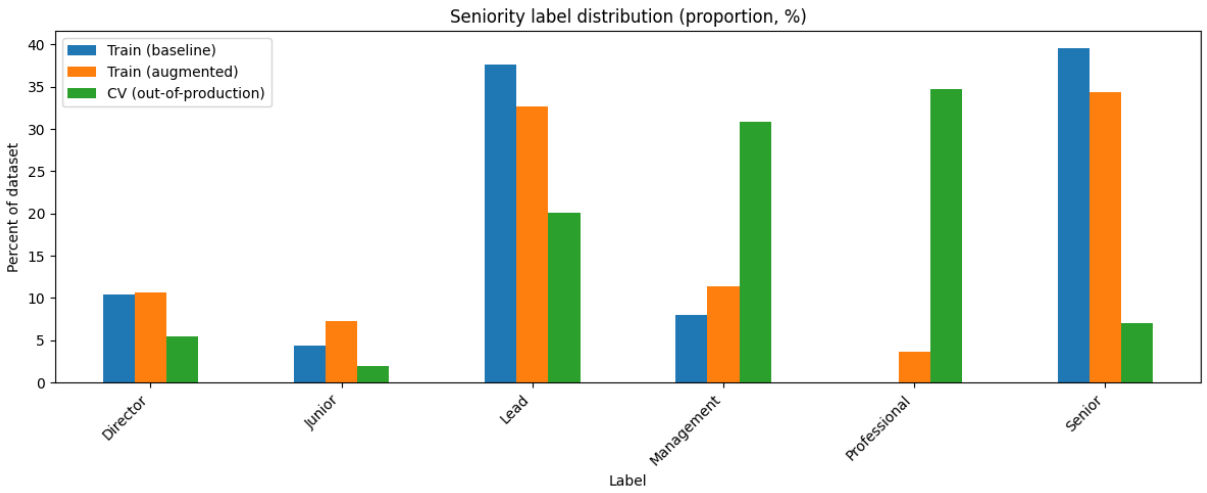

Figure 12: Distribution of seniority labels in different datasets

In Figure 12, the CV dataset is dominated by *Professional*, while the original training data contains almost no *Professional* examples. By adding prompt-labeled synthetic data, we introduce at least a small amount of *Professional* supervision and move the training distribution slightly closer to the CV distribution.

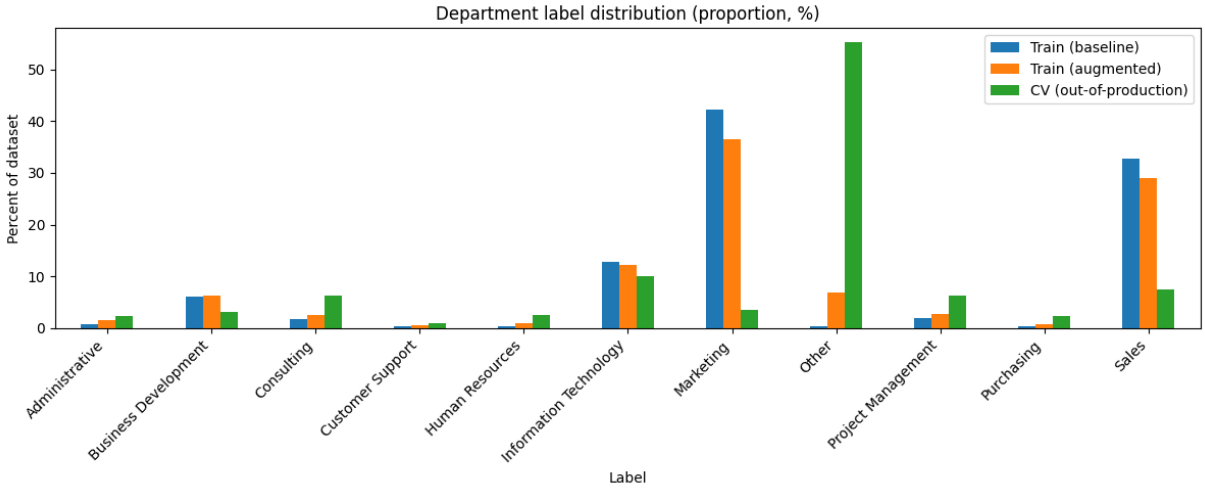

Figure 13: Distribution of department labels in different datasets

In Figure 13, *Other* is the most frequent class in the CV dataset, but is underrepresented in the original training data; synthetic augmentation increases the number of *Other* samples available during fine-tuning, which is expected to help the model learn this class better.

At the same time, we need to account for potential label noise in the synthetic data: in the prompt evaluation, *Business Development* was the weakest department category (lowest F1), so synthetic labels for this class may be less reliable. We therefore treat synthetic augmentation as an empirical experiment and evaluate downstream performance to determine whether the additional data improves robustness on the CV (out-of-production) distribution.

4.4 Fine-tuned classification model

The implementation of our fine-tuning experiments is provided in the GitHub repository under the folder *src/fine_tuning_pretrained/*. We train transformer-based models to predict seniority level and department directly from job titles, and we evaluate both in-distribution performance (on the curated fine-tuning datasets) and out-of-distribution performance on real CV data. Across all experiments we use `xlm-roberta-base`, since job titles in our data are multilingual and, in our preliminary runs, it generalized better to CV data than smaller alternatives (e.g., `distilbert`). For in-distribution evaluation, we split the curated datasets (`df_seniority` and `df_department`) into train/validation/test. Training updates are performed on the train split, early stopping and model selection use the validation split, and we report final in-distribution performance on the held-out test split. For out-of-distribution evaluation, we use `jobs_annotated_df` (real CV job titles) exclusively as a post-training benchmark; it is never used for training or early stopping.

4.4.1 Seniority: fine-tuning approaches and results

We study two seniority fine-tuning modeling strategies, motivated by a strong distribution shift between curated fine-tuning data and real CV job titles (see also the label distribution plots in Figure 12).

1) Regression fine-tuning (no synthetic data, no oversampling). We first map seniority labels to a numeric ordinal scale and fine-tune a regression head. To keep results comparable to classification setups, we additionally report thresholded accuracy/F1 by mapping predicted scores back into label bins. In-distribution performance is very strong: on the test split we obtain MAE = 0.1578 with thresholded accuracy = 0.9929. However, on the annotated CV dataset performance drops substantially to MAE $\approx$ 0.78, indicating that the model does not transfer well to production-like job titles under distribution shift, especially because the CV data contains the label *Professional* while the original fine-tuning dataset does not.

pred_label label	Director	Junior	Lead	Management	Professional	Senior
Director	31	0	1	2	0	0
Junior	0	5	2	0	0	5
Lead	0	3	75	3	2	42
Management	21	1	11	129	6	24
Professional	0	19	23	7	27	140
Senior	1	1	1	0	0	41

Figure 14: Confusion Matrix (All Predictions) – Counts (True label = rows, Predicted = columns)

The confusion matrix (Figure 14) reveals the following insights about our predictions on the CV data:

- **Clear distribution shift:** *Professional* is the most frequent label in CV data but does not exist in the fine-tuning dataset, which explains many downstream confusions.
- ***Professional*** $\rightarrow$ ***Senior/Lead*****:** The model often predicts *Senior* or *Lead* for *Professional* CV titles, consistent with these being the most frequent (and closest) classes seen during training. Misclassifications into *Junior* occur less often, likely because *Junior* is underrepresented in the fine-tuning data.
- **Class imbalance effect:** *Senior* and *Lead* dominate the fine-tuning data, which biases the model toward these labels in ambiguous cases. This motivates using oversam-

pling in the next approach.
- **Rare CV labels:** *Junior* and *Director* are underrepresented in the CV data, making their predictions less stable. We also address this via oversampling in the next step.
- **Consistent error pattern:** Most mistakes occur between adjacent seniority levels, which is expected given the ordinal structure of the labels and the shifted label distribution.

2) Multi-class classification with synthetic data and oversampling To align the label space with the CV setting, we switch to a multi-class classification setup and augment the *training split* with synthetic samples generated via prompt engineering. This adds the previously missing label *Professional* to the training data without using any CV annotations (i.e., without leakage). Since the augmented training data is still imbalanced, we apply oversampling on the training split only, while keeping validation and test unchanged for fair early stopping and model selection.

In-distribution performance on the original test split remains high (accuracy = 0.9611, macro F1 = 0.7926). More importantly, performance on the CV dataset improves substantially compared to the regression baseline, reaching CV accuracy $\approx$ 0.6517 and CV macro F1 $\approx$ 0.5840. Note that *Professional* has zero support in the in-distribution validation/test classification reports by construction: synthetic samples are added only to training, while validation and test remain clean samples from the original dataset.

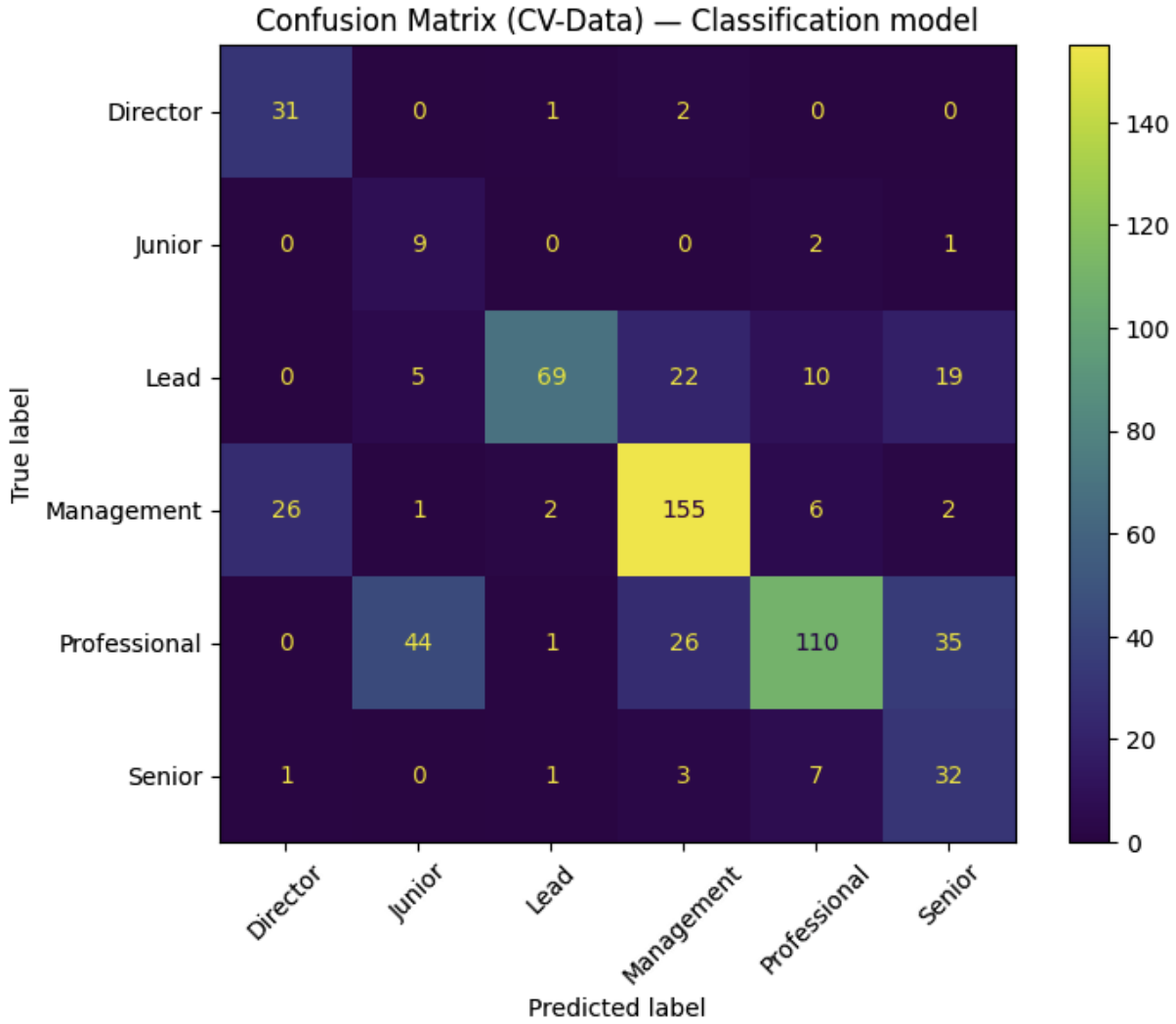

Figure 15: Confusion matrix on the CV dataset after adding synthetic training data and applying oversampling.

Figure 15 shows that the model learns several seniority classes reliably on CV data. *Management* is classified strongly, and *Director* also exhibits comparatively few confusions, which indicates that these categories contain clearer job-title cues. *Junior* is mostly predicted correctly when present, suggesting that explicit junior-level markers are learned effectively.

The remaining errors are concentrated in semantically overlapping or adjacent levels. *Professional* is frequently confused with *Junior*, *Senior*, and *Management*, reflecting that *Professional* is a broad category and often not explicitly expressed in job titles. *Lead* is commonly misclassified as *Senior* or *Management*, which is consistent with ambiguous titles that can describe either technical leadership or people management. Confusions between *Senior* and *Professional* remain common, indicating that the boundary between these classes is

still difficult to learn from job titles alone.

These error patterns likely persist for three reasons: (1) even with synthetic augmentation, *Professional* remains relatively heterogeneous and is still underrepresented compared to its frequency in CV data, (2) many CV job titles do not explicitly encode seniority and would require additional context beyond the title, and (3) synthetic and curated training titles differ in style and noise level from real CV titles, which limits generalization.

Overall, synthetic augmentation plus oversampling improves robustness on real CV job titles (CV accuracy ≈ 0.65, macro F1 ≈ 0.58). However, the dominant remaining failure mode is still ambiguity between neighboring seniority levels, especially around the broad *Professional* category.

4.4.2 Department fine tuning approaches and results

As already mentioned in figure 13, the department fine-tuning dataset is highly imbalanced: most job titles fall into *Marketing* and *Sales*, while classes such as *Human Resources*, *Customer Support*, *Purchasing*, *Administrative*, and especially *Other* are sparsely represented. In contrast, the out-of-production CV dataset follows a markedly different distribution where *Other* dominates and *Marketing* and *Sales* are much less frequent. This mismatch represents a strong distribution shift between training and deployment data. Additionally, department labels are non-ordinal and can overlap semantically (e.g., *Sales* vs. *Business Development* vs. *Consulting*), while *Other* acts as a catch-all category in CV data, increasing ambiguity.

We evaluate three training variants on the same splits: **(1) baseline fine-tuning with department csv data**, **(2) fine-tuning with oversampling on the training split**, and **(3) fine-tuning with synthetic data augmentation**. For each variant we report in-distribution performance on train/validation/test and out-of-distribution performance on the CV dataset, focusing on macro-averaged metrics due to class imbalance.

1) Baseline fine-tuning (no oversampling, no synthetic data). In-distribution results are near-perfect (test accuracy ≈ 0.9980, macro F1 ≈ 0.9913), indicating that the model fits the fine-tuning distribution extremely well. However, performance drops sharply on CV data (accuracy ≈ 0.2793, macro F1 ≈ 0.3813), demonstrating that the learned decision boundaries do not transfer to production-like titles under distribution shift. Figure 16 highlights a systematic failure mode: many CV examples with the true label *Other* are predicted as more specific departments such as *Information Technology* or *Administrative*. This is consistent with *Other* being rare in the training data but dominant in CV data, and with the model relying on training-specific lexical patterns.

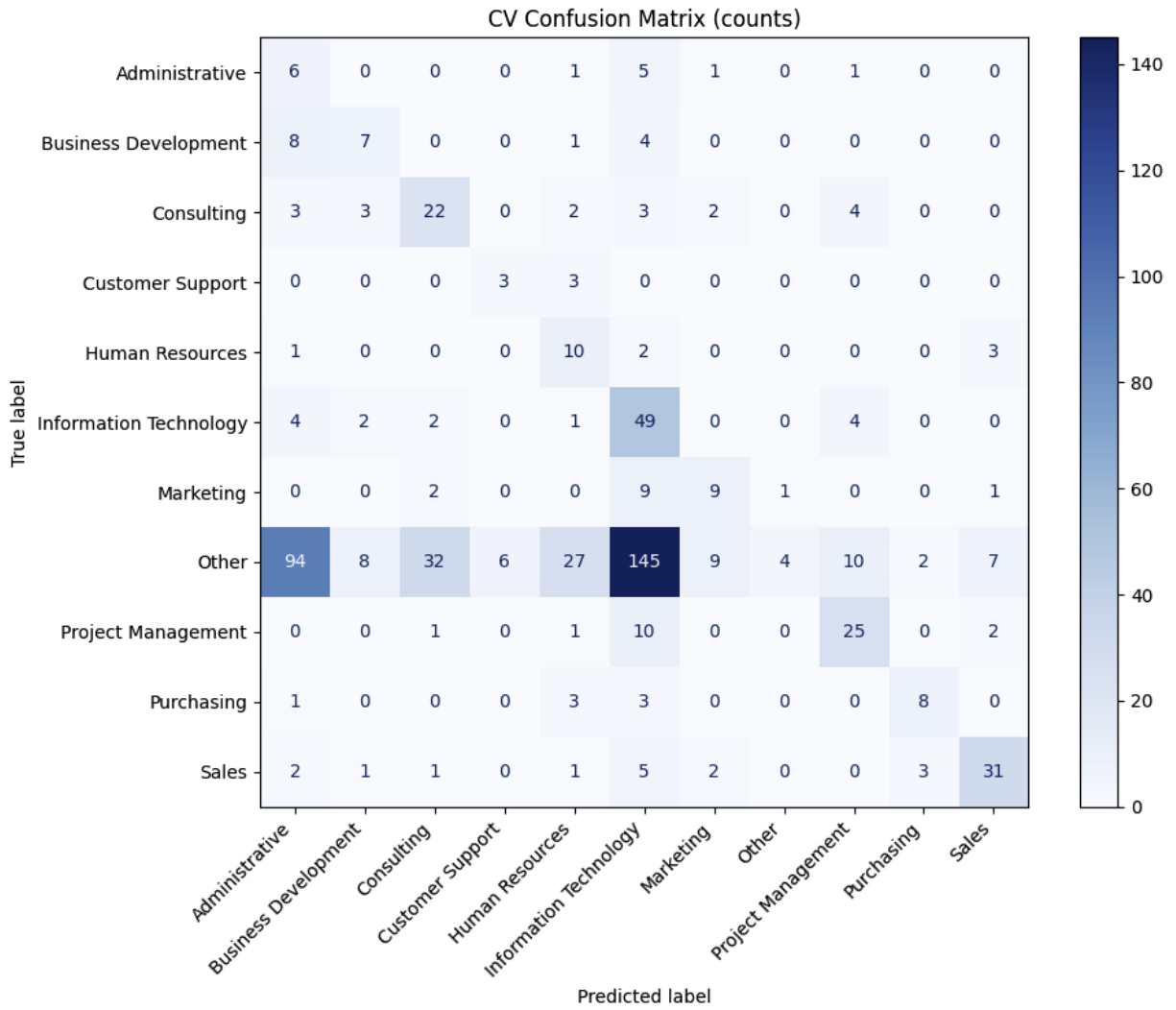

Figure 16: Confusion matrix on the CV dataset for the baseline department classifier.

2) Fine-tuning with oversampling. To mitigate the strong class imbalance, we oversample minority departments in the *training split only*, while keeping validation and test unchanged. This again yields near-perfect in-distribution metrics (test accuracy ≈ 0.9993, macro F1 ≈ 0.9983), but it does not improve robustness on CV data (accuracy ≈ 0.2793, macro F1 ≈ 0.3459). The confusion matrix in Figure 17 shows that the dominant error pattern remains: *Other* examples are still frequently mapped to specific departments. In this setting, oversampling mainly increases exposure to duplicated minority examples from the same distribution, which improves in-distribution fit but does not address the distribution mismatch to CV titles.

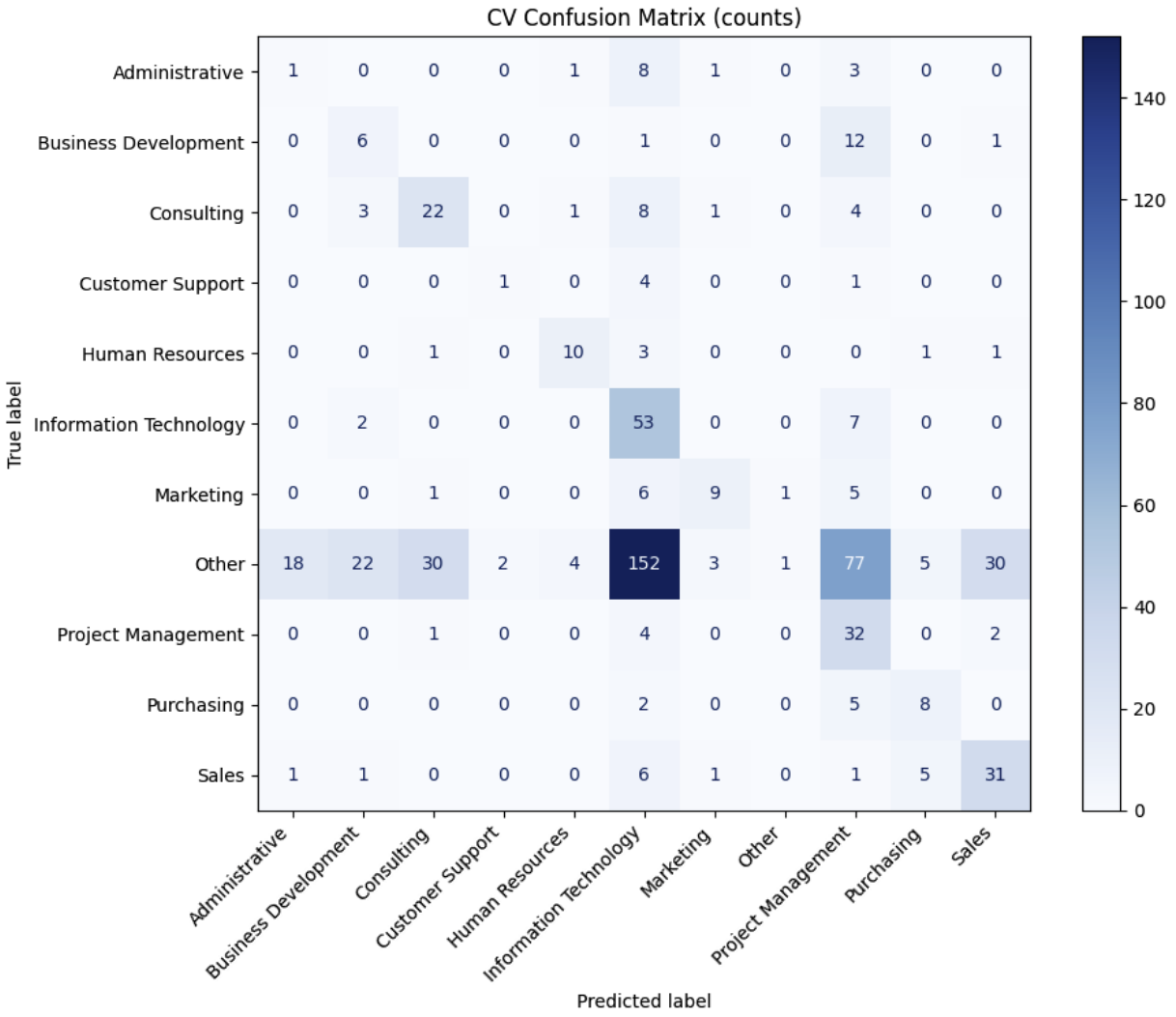

Figure 17: Confusion matrix on the CV dataset for the oversampled department classifier.

3) Fine-tuning with synthetic data augmentation. Finally, we augment the training split with synthetic department labels generated via prompt engineering. The primary goal is to increase both coverage and diversity of job-title formulations, especially for *Other* and other underrepresented departments, thereby reducing the training–CV distribution gap. With synthetic augmentation, in-distribution performance remains strong but decreases compared to the baseline (test accuracy ≈ 0.9947, macro F1 ≈ 0.9770), which is expected because the task becomes harder and the synthetic labels introduce additional variability.

Crucially, out-of-distribution performance on CV data improves substantially (accuracy $\approx$ 0.6886, macro F1 ≈ 0.6374). Figure 18 illustrates why: the model predicts *Other* much more reliably (235 correct *Other* predictions), directly addressing the dominant failure mode of the baseline and oversampling variants. At the same time, the diagonal mass for core

departments (e.g., *Sales*, *Information Technology*, *Project Management*) remains visible, indicating that improved *Other* handling does not simply collapse predictions into a single class. Remaining confusions are primarily between *Other* and *Business Development*.

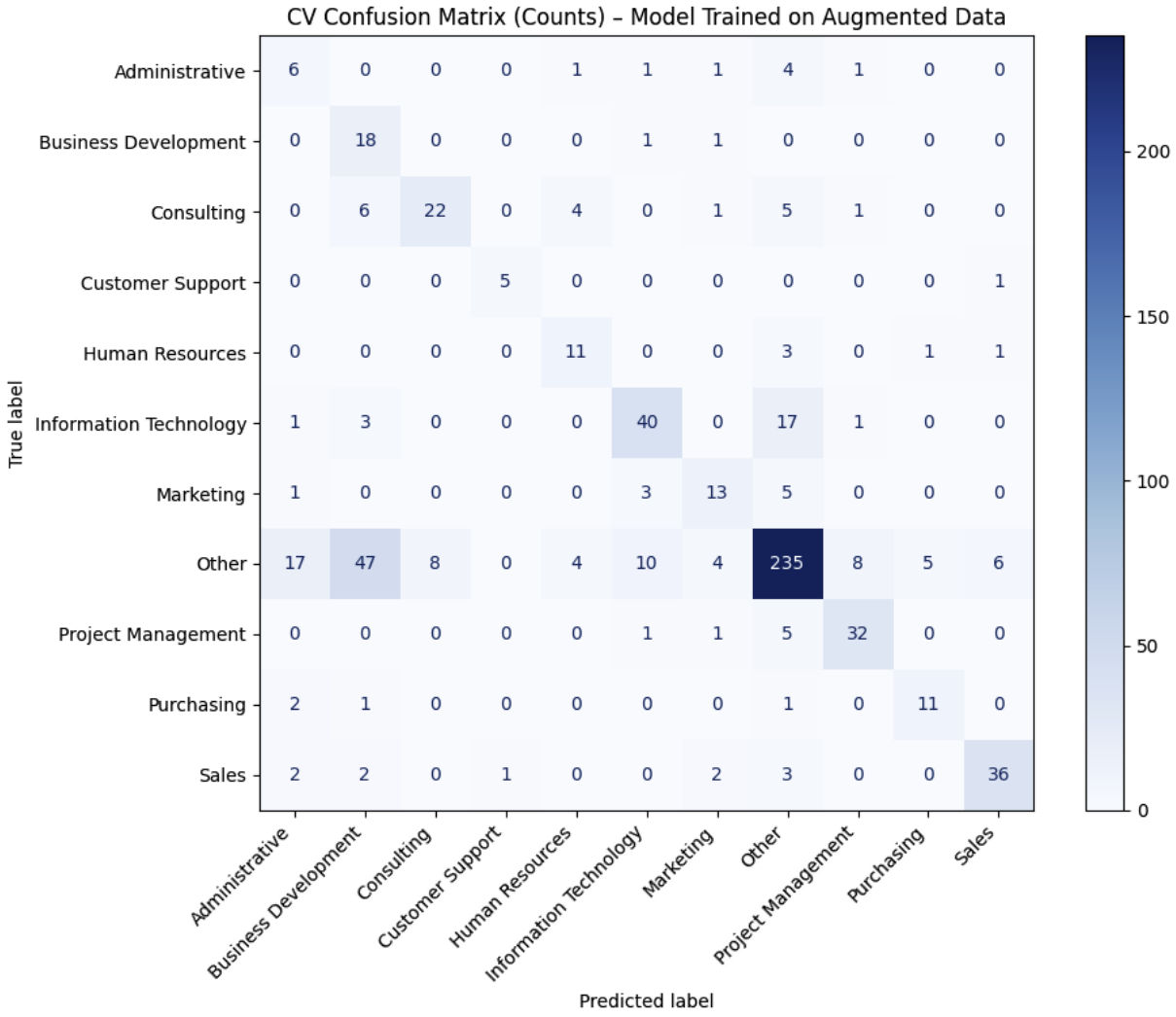

Figure 18: Confusion matrix on the CV dataset after training with synthetic department data.

Across variants, we observe a consistent pattern: the model can achieve extremely high in-distribution scores on the curated dataset, but this does not translate to real CV data due to a pronounced distribution shift, dominated by the *Other* class. Oversampling improves in-distribution balance but does not improve CV robustness. In contrast, synthetic augmentation reduces the distribution mismatch and yields a large gain in out-of-distribution performance, making it the most effective approach for department prediction in the CV setting.

Model variant	Target	Accuracy	Macro F1	MAE
Fine-tuned pretrained	Seniority	0.4943	0.4756	0.7751
Fine-tuned pretrained + synthetic	Seniority	**0.6516**	**0.5840**	–
Fine-tuned pretrained	Department	0.2792	0.3813	–
Fine-tuned pretrained + synthetic	Department	**0.6886**	**0.6374**	–

Table 7: Summary of out-of-distribution results on the annotated CV dataset for the fine-tuned pretrained models. Synthetic augmentation consistently improves robustness under distribution shift for both seniority and department prediction.

4.5 Hybrid Approach (Rule-Based + Fine Tuning

Given the relatively strong performance of the rule-based baseline, we explored a hybrid strategy: we first apply rule-based matching, and only if no rule fires we classify the remaining titles with the fine-tuned model (instead of using a baseline fallback label). The implementation is provided in *src/baseline_hybrid_finetuned_approach.ipynb*.

Seniority. A key limitation of the rule-based system is that it does not cover the label *Professional*. Since *Professional* is present in the CV dataset but missing from the supervised fine-tuning data, we first measure baseline performance in a setting where *Professional* is excluded and no fallback is applied. In this restricted evaluation, the rule-based baseline achieves an accuracy of 73% (300 predictions). This suggests that, when the label space matches the rules, the baseline can be competitive.

Motivated by this, we tested whether adding synthetic data (which includes *Professional*) could make the hybrid approach viable without relying on a fallback. However, when we apply the rule-based baseline with the synthetic augmentation, performance drops to an accuracy of 58% . This is below the performance of the fine-tuned model and indicates that the additional synthetic labels introduce too much noise for this hybrid setup to be beneficial. Therefore, we do not pursue the hybrid approach further for seniority.

Department. We also tested the hybrid idea for department prediction, where the label space is consistent across datasets (i.e., there is no missing label analogous to *Professional*). Nevertheless, the rule-based baseline without fallback achieves only 49% accuracy (204 predictions), which is not competitive. As a result, we also discard the hybrid approach for department prediction.

4.6 Embedding-based labeling

5 Summary of Findings

main findings of eda and label distribution Table of all result metrics -> and which one is the best model and hwy

6 Limitations and Future Work

First, our prompt-engineering setup was only iterated a limited number of times. More systematic prompt iterations could target the most confusing class boundaries directly (e.g., *Business Development* vs. *Sales/Consulting*, and *Administrative* vs. *Other*). This is particularly relevant because *Business Development* was the weakest department class in the prompt evaluation and also remained one of the most difficult categories in the fine-tuned model trained with synthetic data. Improving the prompt in these regions would likely reduce noise in the synthetic labels and therefore improve downstream fine-tuning.

Second, seniority prediction remains strongly constrained by missing or insufficient supervision for the *Professional* label. Even though synthetic augmentation introduces *Professional* examples, the class is still broad and heterogeneous and is a frequent source of confusion on CV data. Future work should therefore prioritize collecting additional high-quality labeled samples for *Professional* (and related borderline cases such as *Professional* vs. *Senior*) to stabilize the decision boundary and improve robustness in production-like settings.

7 Dashboard Implementation

To complement the experimental results, we implemented an interactive dashboard using `Streamlit`. The dashboard is intended as a lightweight presentation and exploration tool rather than a full reproduction of all experiments, as some approaches require GPU resources or offline training and are therefore not suitable for interactive execution.

The dashboard allows users to enter a single job title, representing the current position in a CV, and returns predictions for seniority level, department, and an associated confidence score. Three inference modes are supported: a rule-based baseline using substring matching, a Bag-of-Words model based on TF–IDF features and logistic regression, and a prompt-engineering approach using a large language model (Gemini). The Streamlit entry point is located in `homepage.py`, while all dashboard-specific logic is organized in the `dashboard/` folder. This structure follows Streamlit deployment conventions and cleanly separates the user interface from the modeling components.

The prompt-engineering mode relies on the Google Gemini API and therefore requires a valid API key. For security reasons, the key is not hard-coded but must be provided via a Streamlit secrets variable during deployment, using a file `dashboard/.streamlit/secrets.toml` with the entry

```
GEMINI_API_KEY = "YOUR_API_KEY"
```

During the correction of this report, the API key may be expired, as it was created under a free trial. In this case, clicking the *Predict* button in prompt-engineering mode will not produce a response. Importantly, this behavior does not indicate a coding error: when the *Debug* option in the sidebar is enabled, the dashboard explicitly shows that the request fails due to an invalid or missing API key, allowing the issue to be clearly attributed to authentication rather than implementation.

All dashboard predictions include a confidence value in the range $[0, 1]$, where low values indicate high uncertainty and higher values suggest more reliable predictions. For the rule-based and Bag-of-Words approaches, this score is computed heuristically based on match specificity or model output probabilities, while for prompt engineering it is generated directly by the language model and reflects its own uncertainty assessment. The confidence score is intended as an interpretability aid for demonstration purposes and should not be interpreted as a calibrated probability.

Overall, the dashboard provides an accessible interface to demonstrate the behavior, strengths, and limitations of the different labeling approaches under realistic usage conditions.

8 Appendix

8.1 Code Repository

The complete implementation of our project is available in the following GitHub repository: `https://github.com/luisadosch/Final-Project-snapAddy`.

8.2 Group Member Contributions

In this section we summarize the role of each group member:

1. **Sonia Bronner:** Data overview, data preprocessing and preparation, exploratory data analysis (EDA), and implementation of the rule-based matching baseline.
2. **Laura Hüsam:** Implementation of the bag-of-words approach and the embedding-based labeling approach.
3. **Luisa Dosch:** Implementation of the prompt-engineering pipeline (test-set evaluation and synthetic data generation), fine-tuned classification model experiments (with and without synthetic data), and the hybrid approach (rule-based + fine-tuning).

8.3 Use of Gen-AI

We used GitHub Copilot as a coding assistant during implementation. Since Copilot suggestions are generated interactively inside the IDE and are not stored as a persistent prompt log, we cannot provide a complete, reproducible record of the exact Copilot prompts and outputs used throughout development.

However, we also used ChatGPT. These are the specific prompts we used:

- **Prompt 1:** When my train accuracy is 99% and my test accuracy is 32%, what can I do? I predict departments from job titles (different languages) and use `MODEL_CKPT = "xlm-roberta-base"`.
- **Prompt 2:** "CV Confusion Matrix (counts) – trained on augmented data" Improve this title.
- **Prompt 3:** "Which Hugging Face model would you recommend for classifying multilingual job titles into departments and seniority levels?"
- **Prompt 4:** "Explain the difference between macro F1 and weighted F1."
- **Prompt 5:** At the end I will have a second notebook for department prediction. For both notebooks, I will write a README to make it easier to understand what I did. Can you improve this README and convert it to Markdown: …
- **Prompt 6:** Based on my comments, improve this README (also include the main results at the end as a Markdown table): `# Fine-Tuning Models for Seniority and Department Prediction`
- **Prompt 7:** Improve this text for our report so it is written in correct English: …

List of Tables

List of Figures