Capstone Project: Predicting Career Domain and Seniority from LinkedIn Profiles

Dosch Luisa, Bronner Sonia, Hüsam Laura

January 2025

Contents

1 Data Overview

Hier würde ich vlt bisschen über class imbalance in den train vs test reden

2 Data Preprocessing

an Sonia: haben wir hier überhaupt was machen müssen? ansosnten kannst du das auch weglassen

3 EDA

4 Experiments

4.1 Rule Based Matching

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 1 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 1: 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 1 and 2 illustrate why this augmentation is relevant: the label distributions differ substantially between the training data and the CV (out-of-production) dataset.

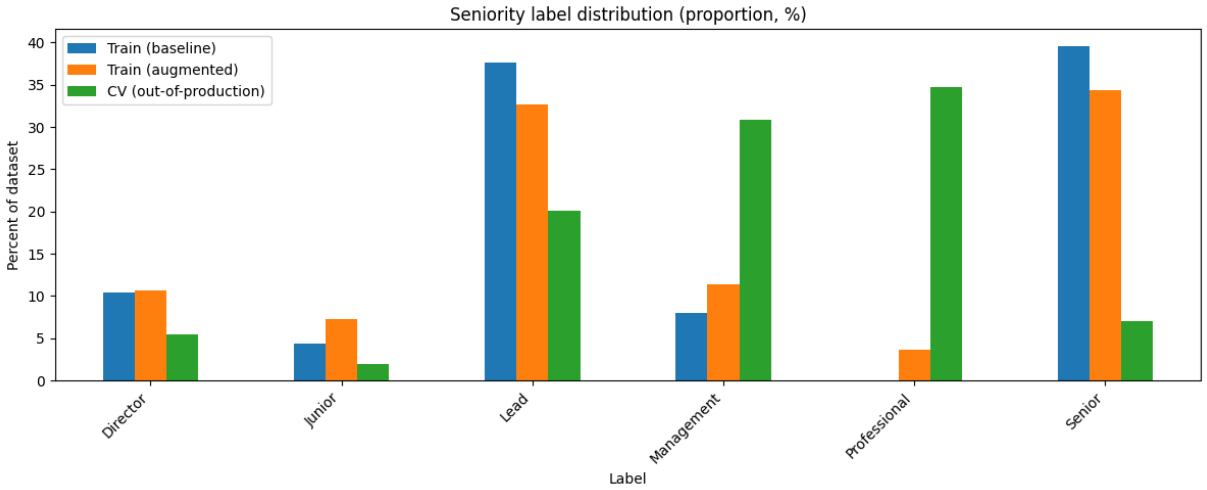

Figure 1: Distribution of seniority labels in different datasets

In Figure 1, 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.

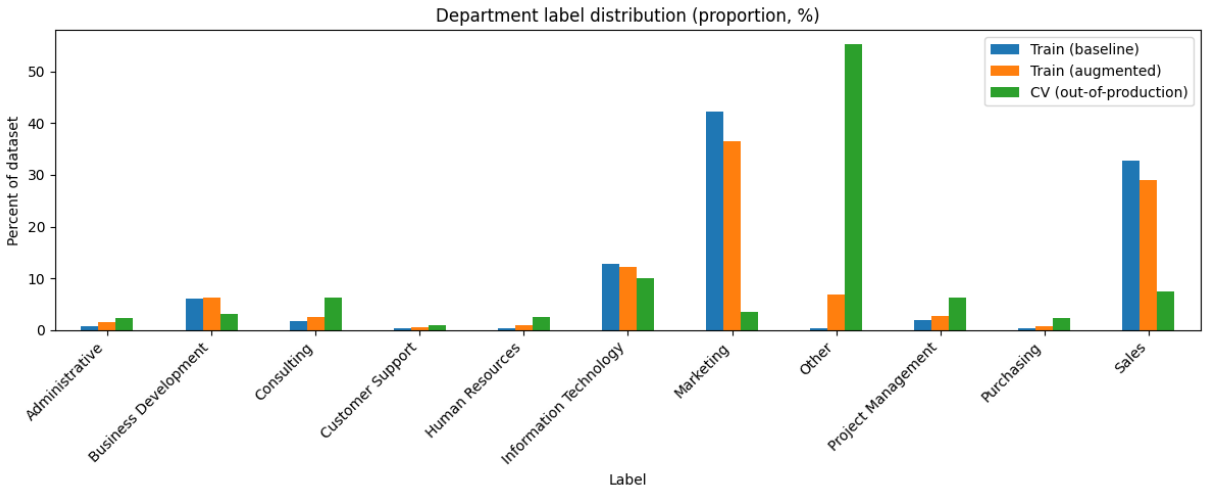

Figure 2: Distribution of department labels in different datasets

In Figure 2, *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 1).

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

The confusion matrix (Figure 3) 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 oversampling 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.

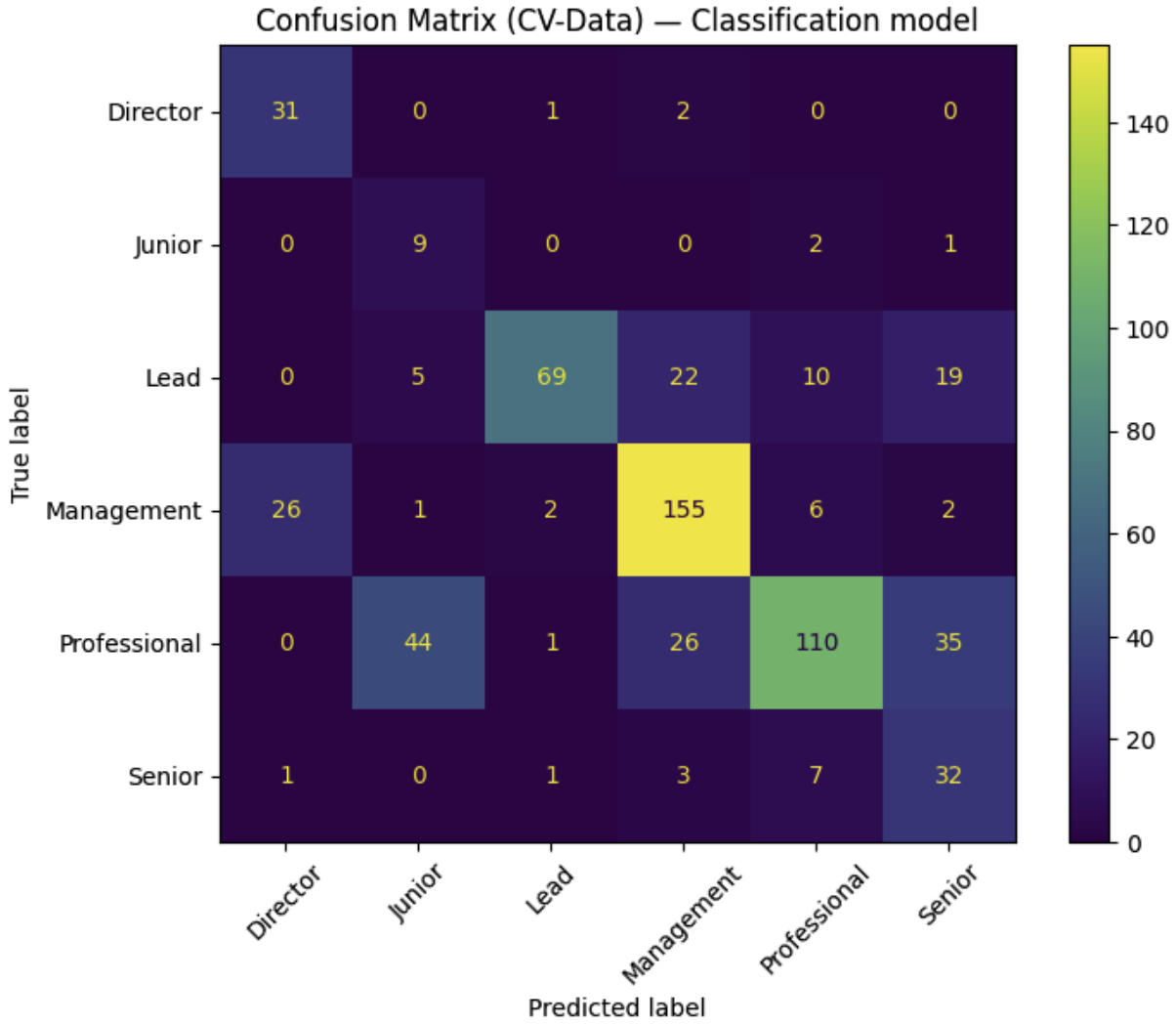

Figure 4: Confusion matrix on the CV dataset after adding synthetic training data and applying oversampling.

Figure 4 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 2, 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 5 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.

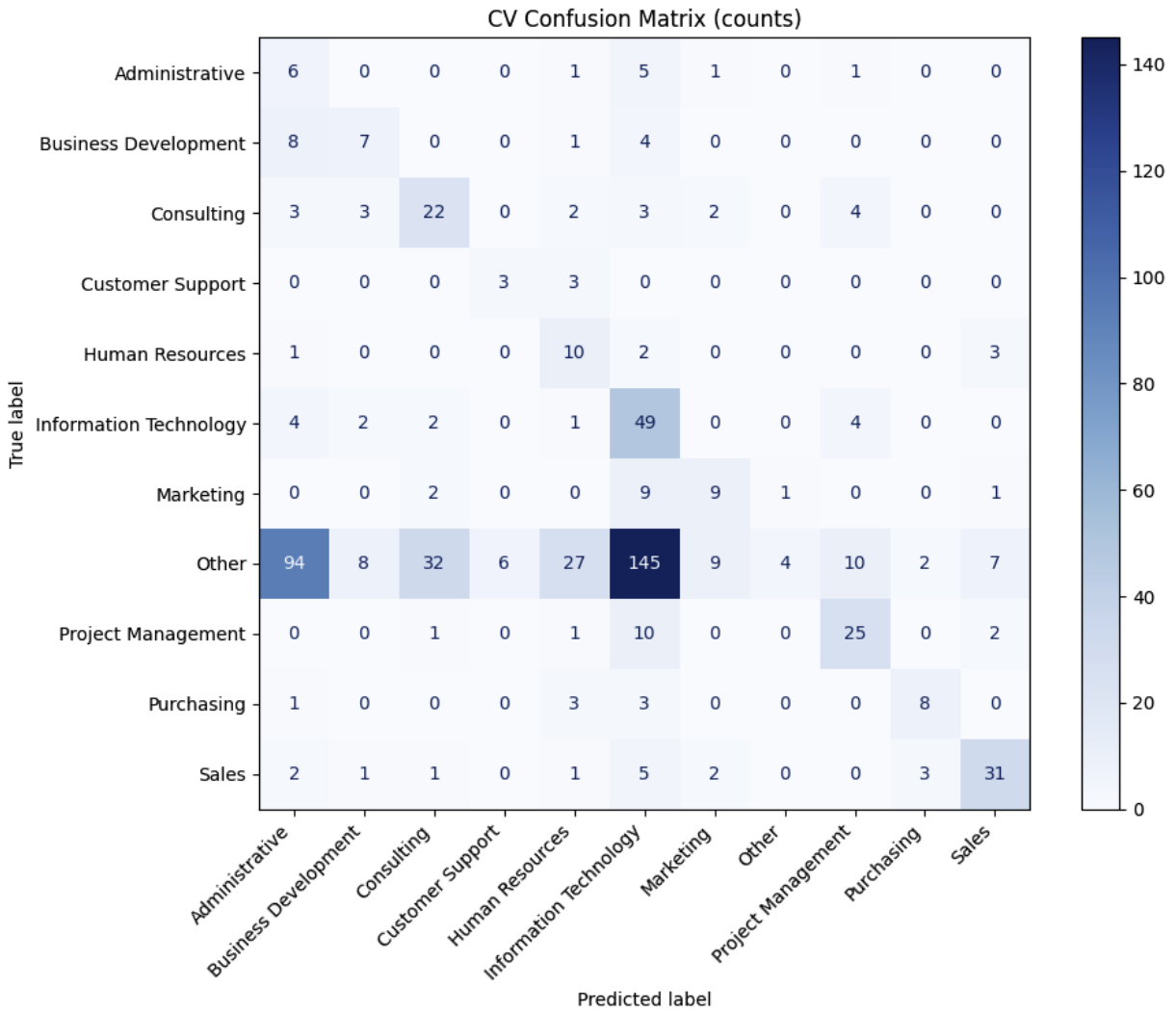

Figure 5: Confusion matrix on the CV dataset for the baseline department classifier.

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 6 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.

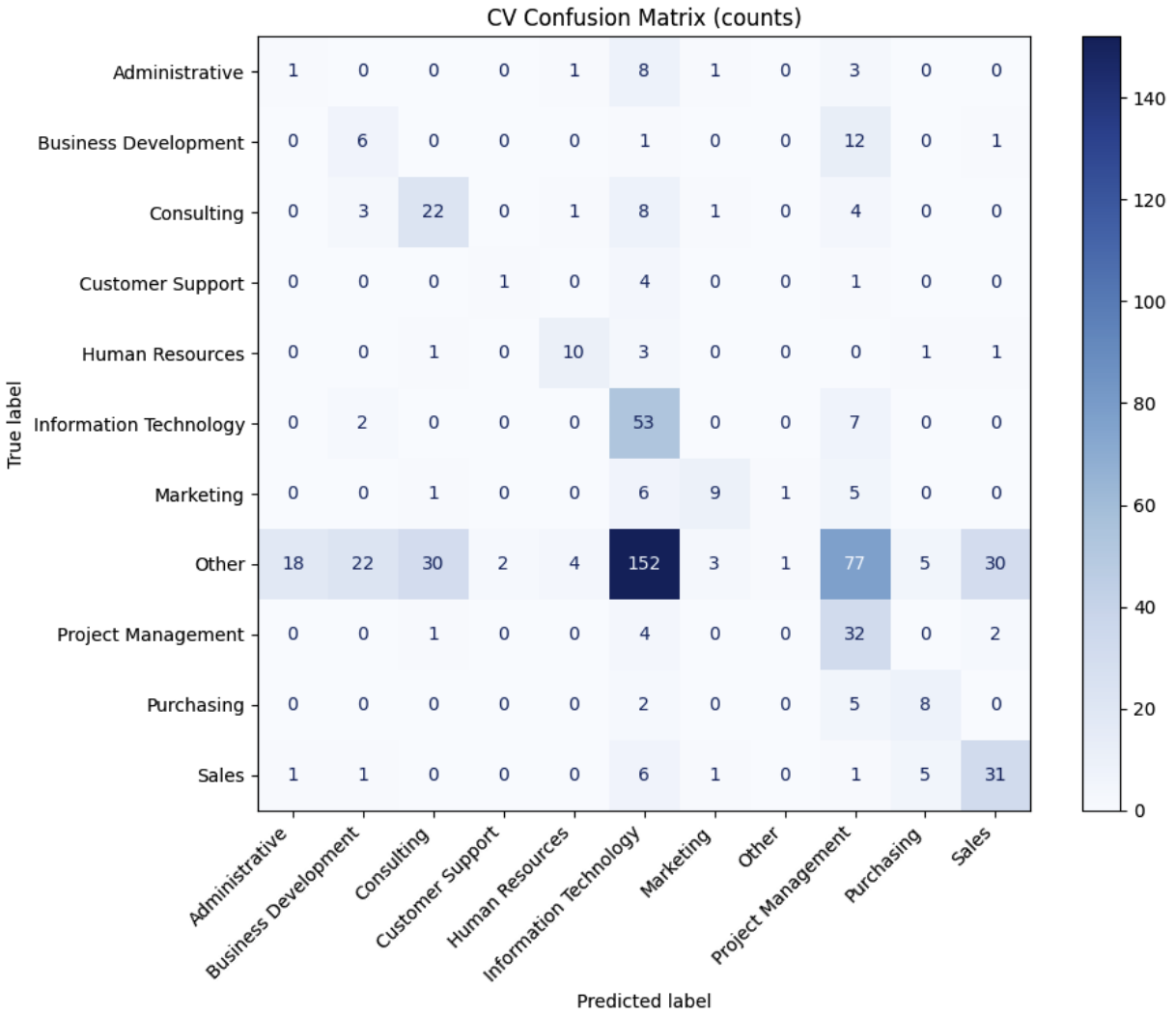

Figure 6: Confusion matrix on the CV dataset for the oversampled department classifier.

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 7 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*.

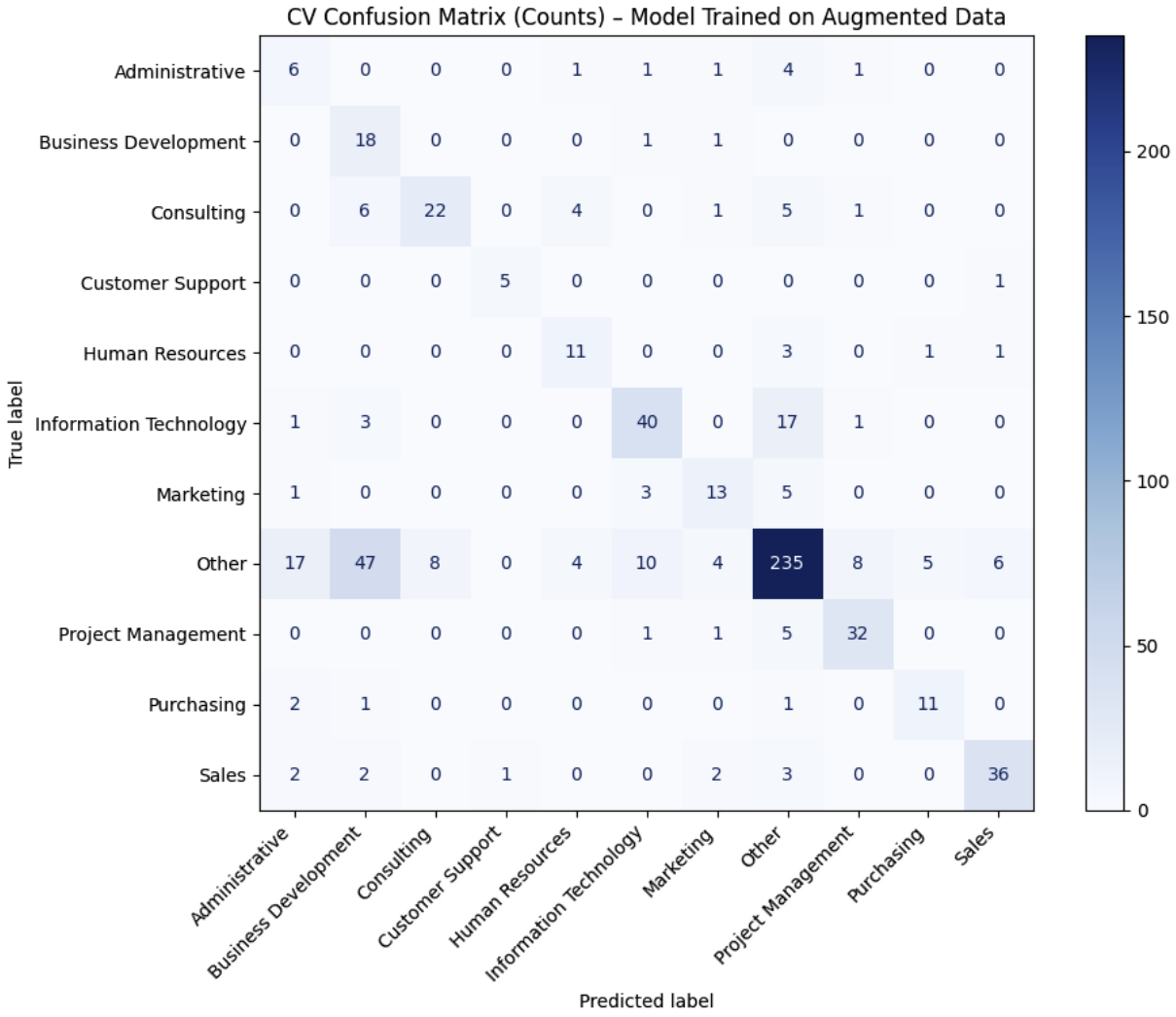

Figure 7: Confusion matrix on the CV dataset after training with synthetic department data.

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 2: 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 Appendix

7.1 Group Member Contributions

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

1. **Sonia Bronner:** Data overview, data preprocessing, 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).

7.2 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.

List of Tables

List of Figures