Building Data-Driven Occupation Taxonomies: A Bottom-Up Multi-Stage Approach via Semantic Clustering and Multi-Agent Collaboration

Nan Li[1], **Bo Kang**[1,2], **Tijl De Bie**[1],
[1]IDLab, Department of Electronics and Information Systems,
Ghent University, Belgium
[2]Nobl.ai
Correspondence: nan.li@ugent.be

Abstract

Creating robust occupation taxonomies, vital for applications ranging from job recommendation to labor market intelligence, is challenging. Manual curation is slow, while existing automated methods are either not adaptive to dynamic regional markets (top-down) or struggle to build coherent hierarchies from noisy data (bottom-up). We introduce CLIMB (CLusterIng-based Multi-agent taxonomy Builder), a framework that fully automates the creation of high-quality, data-driven taxonomies from raw job postings. CLIMB uses global semantic clustering to distill core occupations, then employs a reflection-based multi-agent system to iteratively build a coherent hierarchy. On three diverse, real-world datasets, we show that CLIMB produces taxonomies that are more coherent and scalable than existing methods and successfully capture unique regional characteristics. We release our code and datasets at https://github.com/aida-ugent/CLIMB.

1 Introduction

A taxonomy is a hierarchical structure that organizes information. In the labor market, a robust taxonomy is critical for organizing job postings, guiding job seekers, and informing policy. It improves search and guidance for job seekers, provides governments a framework to analyze labor trends and inform policy, and underpins corporate workforce planning and skill gap analysis. However, because labor markets are dynamic and regionally diverse, there is a strong need for taxonomies that are tailored to specific markets and easily updatable. The goal of this work is to fully automate the construction of such data-driven, hierarchical taxonomies directly from a raw corpus of job postings. While the optimal level of granularity for a taxonomy is application-specific, its quality can be assessed on universal properties such as coherence, comprehensiveness, and efficiency, which our evaluation framework is designed to measure.

Existing methods struggle to meet this need. Manual curation is slow, expensive, and unscalable. As detailed in Section 2, automated approaches are either *top-down*, expanding an existing structure, or *bottom-up*, building a new one from data.

Many traditional and recent LLM-based methods follow a top-down approach. These systems are designed to expand upon a pre-existing *seed*, typically a small, expert-curated taxonomy or a list of initial terms extracted and filtered using classic natural language processing techniques. While effective for enriching existing knowledge structures, this reliance on a seed makes them ill-suited for building a taxonomy from scratch in a new domain and and fails to fully remove the need for expert knowledge and potential for human bias.

In contrast, bottom-up approaches aim to build a hierarchy directly from a raw corpus but face fundamental challenges. While LLMs used in these methods have inherent priors, a truly bottom-up approach ensures that the final taxonomy's structure is dictated by the corpus itself, not by predefined seeds or expert-curated lists. *First*, they struggle to distill a globally consistent set of core concepts (e.g., distinct occupations) from the data; feeding an entire corpus to an LLM is often infeasible, while incremental processing sacrifices the global perspective of the whole corpus needed to identify concepts consistently. *Second*, even with a clean set of concepts, constructing a deep and logically coherent hierarchy is a complex reasoning task where single-LLM systems often fail, producing inconsistent groupings and flawed structures.

To address these challenges, we propose CLIMB (**CL**uster**I**ng-based **M**ulti-agent taxonomy **B**uilder), a multi-stage framework that solves both problems in sequence. To overcome the first challenge, CLIMB begins with the raw corpus and uses semantic clustering over the entire dataset to au-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1596–1614
November 4-9, 2025

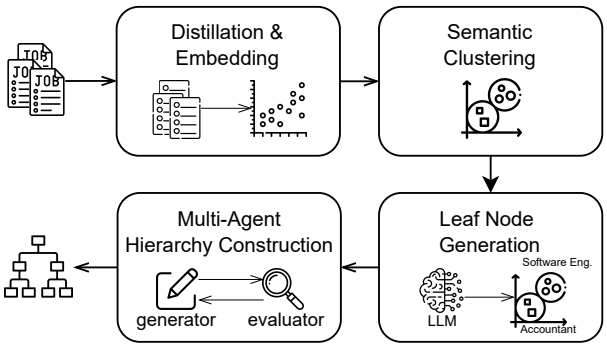


Figure 1: High-level overview of CLIMB.

tomatically distill a globally-informed set of leaf-node concepts. To solve the second, CLIMB employs a novel reflection-based multi-agent framework for the hierarchical construction. It proceeds level-by-level, using a "Generator" to propose parent concepts and an "Evaluator" to critique them for logical consistency. This iterative refinement is essential for building a complex yet coherent taxonomy. Our *contributions* are:

- We propose a novel system design, CLIMB, that fully automates bottom-up taxonomy generation. Its key innovation is a two-stage architecture that combines (1) a global clustering sub-system to distill concepts robustly from a raw corpus and (2) a multi-agent, reflection-based reasoning framework to solve the complex task of hierarchical construction. This design makes the process objective by removing the need for expert seeds or manual curation.
- We demonstrate that CLIMB produces taxonomies that are not only coherent and scalable, but also highly adaptive. Unlike static, generic taxonomies, CLIMB's data-driven approach creates structures that are customized to specific regional labor markets at a specific point in time, capturing unique local roles and emerging trends.
- We release our code and the datasets used in this study to the public to encourage reproducible research and further innovation in this area (`https://github.com/aida-ugent/CLIMB`).

2 Related Work

The task of taxonomy generation is broadly divided into two main methods: top-down and bottom-up.

Top-Down Approaches. A significant body of work, from both the pre-LLM era (Shen et al., 2018; Zhang et al., 2018; Huang et al., 2020; Shang et al., 2020; Lee et al., 2022; Le et al., 2023) and the recent LLM era (Zeng et al., 2024; Gunn et al., 2024; Marchenko and Dvoichenkov, 2024; Kargupta et al., 2025), operates top-down by expanding a pre-existing seed taxonomy. While effective for enriching established knowledge structures, this reliance on a seed makes them less suitable for creating a taxonomy from scratch in a new domain.

Bottom-Up Approaches. In contrast, bottom-up approaches aim to construct a taxonomy directly from a corpus, aligning with CLIMB's objective of creating data-driven, adaptive taxonomies. Recent works have explored several strategies to this end. Some domain-specific methods require a pre-compiled list of terms (Sas and Capiluppi, 2024; Moraes et al., 2024), precluding full automation from raw text. To the best of our knowledge, the state-of-the-art method that operates in a fully automated, bottom-up, and seed-free manner is TnT-LLM (Wan et al., 2024), which we select as our primary baseline. TnT processes data in small batches to iteratively build its taxonomy. While scalable, this local view (i.e., seeing only the data in the current batch) can result in redundant concepts or a fragmented hierarchy, especially with noisy data. Another strategy uses a human-in-the-loop to refine an LLM-generated draft, sacrificing full automation for quality control (Shah et al., 2023). In contrast, CLIMB's fully automated solution resolves the local-view limitation by performing global clustering on the corpus, ensuring its foundation of leaf nodes is comprehensive and consistent.

Reflection for Complex Reasoning. Once base concepts are identified, constructing a logically sound hierarchy is a complex reasoning task where a single LLM struggles to ensure overall logical coherence. In other fields require such reasoning, such as code generation (Madaan et al., 2023; Chen et al., 2023) or mathematical problem-solving (Yao et al., 2023), reflection-based multi-agent frameworks using a "generator" and "evaluator" for iterative refinement have proven highly effective (Anthropic, 2024; Kalyanpur et al., 2024; Yuan and Xie, 2025). To our knowledge, CLIMB is the *first* to apply this powerful paradigm to coherent and robust taxonomy construction.

3 Method

Given a corpus of documents, our goal is to automatically construct a hierarchical taxonomy. As aforementioned, building a taxonomy from a raw corpus has two challenges: (1) distilling a consistent set of base concepts from a large and noisy

corpus, and (2) constructing a coherent hierarchy from them. CLIMB's multi-stage pipeline (Figure 1) is designed to address these. *First*, the pipeline generates leaf nodes from raw job postings. This involves optional text distillation and embedding (Section 3.1), global semantic clustering to identify occupations (Section 3.2), and then transforming raw clusters into canonical leaf nodes via LLM-based abstraction and normalization (Section 3.3). *Second*, for hierarchical construction (Section 3.4), a reflection-based multi-agent framework iteratively builds a logically coherent taxonomy upwards from the leaves, level by level. Implementation details are provided in Appendix C.

3.1 Job Posting Distillation & Embedding

Job postings often mix core occupational details with irrelevant text about company culture or application procedures. Applying a large LLM to summarize an entire corpus is prohibitively slow and expensive. We therefore introduce an optional, scalable distillation step: an LLM cost-effectively generates a high-quality training set for a lightweight classifier, which then extracts relevant text segments from all postings for subsequent embedding. The process involves the following key steps:

Data Preparation: All job descriptions first undergo basic text cleaning (e.g., removing HTML tags and extra whitespace) and are then segmented into text chunks for relevance classification.

LLM Labeling: To create training data cost-effectively, an LLM annotates a sample of chunks as relevant or irrelevant to the core occupation.

Classifier Training: A lightweight binary classifier is trained on the LLM-annotated data, enabling scalable filtering of all job postings.

Distilled Description Generation: The classifier extracts relevant chunks from each posting to create a "distilled" description. For robustness, if distillation is not needed or yields no relevant text, the full, preprocessed posting is used as a fallback.

Embedding: The resulting descriptions are embedded using a pre-trained language model for the subsequent clustering stage.

3.2 Semantic Clustering

The next step is to group distilled job descriptions into fine-grained occupation clusters (leaf nodes of a taxonomy tree). Standard clustering approaches using generic embeddings and cosine similarity are insufficient for this task, as they fail to capture the nuanced human judgment of what constitutes the "same occupation" in practice. We therefore develop a specialized, multi-step sub-system: we learn a custom similarity metric by training a classifier to mimic an HR expert's assessment using contrastive data sampling and rich feature engineering. This non-trivial approach is a key contribution that enables the high quality of our leaf nodes. The process involves the following steps:

Contrastive Data Sampling: To create a rich training set, we sample pairs using a coarse initial signal from general-purpose embedding cosine similarity. This is *not* a labeling step, but a strategy to deliberately select a challenging mix for LLM annotation: likely easy-positives (high similarity), likely easy-negatives (low similarity), and likely ambiguous hard-positives and -negatives (moderate similarity).

LLM as HR Expert: We employ a powerful, general-purpose LLM to act as a proxy for an HR expert, a common and proven effective strategy for scalable data annotation(Gilardi et al., 2023; Zheng et al., 2023), to provide ground-truth "same occupation" labels for the sampled pairs. This supervision forces the classifier to learn from these ambiguous pairs and develop a nuanced similarity metric that surpasses the initial embeddings. The prompts are available in our public repository.

Classifier Training: An XGBoost classifier is trained on these LLM-generated labels. The feature set for a pair of job embeddings, e_a and e_b, is the concatenation of $(e_a, e_b, e_a - e_b, e_a \odot e_b)$, capturing rich interaction information.

Clustering with Learned Similarity: The trained classifier is used to compute a similarity score for all job pairs. These scores then serve as the input for the Affinity Propagation algorithm (selected in early tests for its superior silhouette scores), to group postings into distinct occupation clusters.

3.3 Leaf Node Generation

The raw clusters from the previous step are groups of job postings, but their varied and often inconsistent titles are unsuitable for a formal taxonomy. This stage transforms these clusters into clean, canonical occupations with clear titles and descriptions, serving as the leaf nodes of the taxonomy.

LLM-based Abstraction: We prompt an LLM with the cluster's (sampled if too many) job postings to generate a concise title and description for the occupation each cluster represents, crucial for abstracting a canonical concept from noisy raw text.

Normalization and Deduplication: LLM-generated titles can be ambiguous or redundant, so we refine

them as follows. (1) Ambiguous conjoined titles (e.g., "Accountant and Bookkeeper") are removed to ensure one concept per node. (2) Standard text cleaning is applied. (3) Semantically equivalent nodes are merged by clustering their embeddings with Affinity Propagation, which identifies an exemplar title for each group.

This process of abstraction and normalization ensures our leaf nodes are distinct and well-defined, forming a robust foundation for the hierarchy.

3.4 Hierarchical Taxonomy Construction

With a solid set of leaf nodes, we build the taxonomy upwards. This is a complex reasoning task: grouping concepts from specific to general into a coherent hierarchy, similar to standard taxonomies. Single-pass LLM approaches are known to fail at such complex reasoning tasks, prone to errors that propagate and corrupt the entire tree. To address this, we employ a novel application of the reflection-based multi-agent paradigm to taxonomy construction: a deliberate, level-by-level framework with a "Generator" and an "Evaluator" to ensure coherence at each stage of construction. Starting with the leaf nodes (Level 0), the hierarchy is built using the following iterative process.

Generator: An LLM proposes parent concepts to group the current level's nodes. This step performs the specific-to-general reasoning, outputting parent titles, descriptions, and child-parent mappings.

Evaluator: To prevent error propagation, an Evaluator agent scrutinizes the Generator's output for logical consistency. It checks for common failure modes identified in our early explorations: missing child nodes, children assigned to multiple parents, and hallucinated mappings to non-existent nodes.

Generate-Evaluate Cycle: If the Evaluator finds flaws, it provides feedback to the Generator to refine its output. This cycle repeats until the structure is validated. The new parent nodes then become the input for the next level, a process that continues until the number of newly generated parent nodes falls below a set threshold.

4 Experiments

To evaluate CLIMB's performance, we use three real-world datasets from different regions with varied sizes, comparing it against two baselines using a variety of evaluation metrics.

Datasets. We collected three datasets from major job search websites in their respective regions: **Palestine** with 2701 postings in English and Arabic, **Botswana** with 4854 English postings, and **USA** with 11285 English postings. For each dataset, we randomly split the data into training and testing sets with a ratio of 9:1. The training set is used for generating the taxonomy, and the testing set is used for evaluating the quality of the taxonomy. More details are provided in Appendix A.

Baselines. We compare CLIMB against a state-of-the-art automated method and an expert-curated standard. **TnT** (Wan et al., 2024): As introduced in Section 2, TnT is our primary baseline as the state-of-the-art fully automated, bottom-up method. As its original prompts create a flat structure for user intents, we adapted them for the occupation domain. **TnT-H**: To provide a stronger hierarchical comparison, we created TnT-H, a hierarchical variant of TnT. This required substantial re-architecture: a recursive, multi-pass process that generates top-level categories, assigns documents to categories, and recursively applies TnT to each subset. We evaluate this computationally expensive baseline on the Palestine dataset. **ESCO** (European Commission, 2024): Built upon the international ISCO-08 (International Labour Organization, 2008) standard, the European Skills, Competences, Qualifications and Occupations (ESCO) is a comprehensive, expert-curated taxonomy designed for *region-agnostic* application. As the full taxonomy is too large to fit within the context window of the LLM annotators used for evaluation, we use a two-level version corresponding to its four- and five-digit codes, a necessary simplification.

Evaluation Metrics. To evaluate the taxonomies scalably, we employ a panel of three LLMs as independent annotators, a common and cost-effective strategy for reliable annotation (Gilardi et al., 2023; Zheng et al., 2023). Our panel consists of Gemini 2.5 Flash, O4-Mini, and DeepSeek R1 (see Appendix C for full model details). A detailed analysis of the annotator panel's reliability and bias is provided in Appendix E. These annotators are tasked with labeling job postings from the test set using a given taxonomy. We then assess the quality of each taxonomy across three key dimensions: accuracy, comprehensiveness, and efficiency, using standard metrics from prior work (Shah et al., 2023; Wan et al., 2024) applied to the LLM-generated labels:

Accuracy. High-quality taxonomies should be consistent (no contradictions), clear (unambiguous definitions), and accurate (correct categorizations).

Without the ground truth labels for the job postings' occupations, we use the inter-annotator agreement as a proxy for label accuracy. For hierarchical taxonomies, we also measure the hierarchical agreement to provide a more detailed view. This gives us the following two metrics. **Strict Agreement:** Requires all annotators to assign identical occupation labels to a job, and not choose "Other." This strict metric evaluates label clarity and consistent interpretation. **Hierarchical Agreement:** A more flexible metric for tree structures where agreement is met if assigned labels share an identical parent node at a given level. This rewards semantically close, if not exact, predictions. We use *Fleiss' kappa* (Fleiss, 1971) to measure agreement, interpreting values using standard ranges (Landis and Koch, 1977; McHugh, 2012): < 0.20 (slight), $0.21-0.40$ (fair), $0.41-0.60$ (moderate), $0.61-0.80$ (substantial), $0.81-1.00$ (almost perfect).

Comprehensiveness. Measured by **Coverage Rate**, the percentage of jobs assigned a specific label rather than "Other." A high rate indicates the taxonomy is sufficiently comprehensive for the corpus.

Efficiency. Measured by **Label Utilization Rate**, the percentage of labels used at least once during annotation, indicating how much of the taxonomy's breadth is relevant.

Table 1: Comparison of taxonomy sizes (number of nodes) against training dataset sizes. L0, L1, L2 refer to the levels of the taxonomy, from specific to general; '-' indicates a non-existent level. ESCO is a dataset-agnostic static taxonomy. TnT generates lower number of nodes in general, and does not appear to scale with the size of the training dataset. CLIMB generates taxonomies that scale with the size of the training dataset.

Dataset (# Jobs)	Taxonomy	L2	L1	L0	Total
-	ESCO	-	436	1760	2196
Palestine (2430)	TnT	-	-	87	87
	TnT-H	15	56	72	143
	CLIMB	14	32	91	143
Botswana (4368)	TnT	-	-	200	200
	CLIMB	29	99	277	418
USA (10129)	TnT	-	-	130	130
	CLIMB	190	341	671	1298

5 Results

We present quantitative (Section 5.1) and qualitative (Section 5.2) results.

5.1 Overall Performance & Analysis

We present our main quantitative results in this section. Table 1 compares the sizes of the generated taxonomies against the data sizes used for generating them. For each dataset, we report the number of nodes at the levels that are used later for evaluation, where the total number of nodes is the sum of the nodes at all levels. The interactive visualizations of the trees generated by CLIMB are online `https://shorturl.at/L0kpT`. More descriptions are provided in Appendix D.

Table 2: Taxonomy comparison across datasets (agreement numbers are kappa values with interpretation symbols, others are percentages). Acronyms: Agr. (Agreement), Util. (average Label Utilization), Cov. (average Coverage Rate). Kappa interpretation: *= substantial, **= almost perfect. †Agreement and Utilization at L2 for a fairer comparison between CLIMB and TnT where node numbers are similar (CLIMB: 190 vs. TnT: 130).

Dataset	Taxonomy	Strict Agr.	Util.	Cov.
Palestine	ESCO	0.46	5.24	95.33
	TnT	0.59	48.66	89.30
	TnT-H	0.60	61.77	94.34
	CLIMB	**0.73***	**54.50**	**98.52**
Botswana	ESCO	0.56	10.02	**98.83**
	TnT	0.42	38.00	81.21
	CLIMB	**0.63***	**44.86**	97.81
USA	ESCO	0.46	17.28	97.20
	TnT	0.62*	**82.56**	79.48
	CLIMB	0.54	34.58	**97.34**
	CLIMB (L2)†	**0.67***	82.28	97.34

Table 3: Hierarchical agreement for CLIMB (kappa values), with levels numbered from most specific (L0) to most general. Kappa interpretation: *= substantial, **= almost perfect.

Dataset	Level 0	Level 1	Level 2
Palestine	0.73*	0.81**	0.82**
Botswana	0.63*	0.72*	0.76*
USA	0.54	0.66*	0.67*

Table 4: Per-level comparison between CLIMB and TnT-H on Palestine dataset. Both have 143 total nodes but different structures. CLIMB achieves superior agreement at every level, with particularly strong performance at upper levels (L1, L2). TnT-H's higher overall utilization is a misleading artifact of its poorly defined upper hierarchy (only 33.33% of L2 nodes used).

Level	Method	Node Count	Strict Agr.	Util. (%)
L2 (Top)	CLIMB	14	0.82**	**97.62**
	TnT-H	15	0.67*	33.33
L1 (Mid)	CLIMB	32	0.81**	**92.71**
	TnT-H	56	0.63*	55.95
L0 (Leaves)	CLIMB	91	**0.73***	**72.89**
	TnT-H	72	0.60	71.76
Overall	CLIMB	143	**0.73***	54.50
	TnT-H	143	0.60	61.77

Table 2 evaluates the overall quality of the taxonomies using several metrics. To ensure a fair

comparison on the USA dataset, this table includes an additional entry comparing taxonomies of similar size. Table 3 assesses hierarchical agreement at different levels of the taxonomy generated by CLIMB, numbered bottom-up (Level 0 = leaves).

Our main results demonstrate the effectiveness of CLIMB across multiple datasets.

CLIMB generates significantly more consistent and unambiguous taxonomies, leading to higher inter-annotator agreement. As shown in Table 2, CLIMB scores the highest in the Palestine and Botswana datasets with substantial agreement. While TnT appears to have a higher agreement score on the USA dataset, this direct comparison is misleading due to the vast taxonomy size difference (see Table 1). For a fairer comparison, we report CLIMB's agreement at Level 2, where the number of nodes (190) is much closer to TnT's (130). As shown in the last row for the USA dataset in Table 2, CLIMB's 0.67 agreement score surpasses TnT's 0.62, demonstrating superior structure, though its high granularity on this complex dataset reduces leaf-level clarity (Table 3).

The hierarchy produced by CLIMB is logically structured and semantically coherent. As shown in Table 3, agreement scores consistently increase as we ascend the hierarchy from more specific to broader categories, reaching almost perfect agreement for the Palestine dataset. This trend confirms that the generated structure is semantically coherent, as annotators tend to agree on broader concepts even when they diverge on finer-grained distinctions.

CLIMB's multi-agent framework produces superior hierarchical quality compared to recursive generation approaches. To provide a stronger hierarchical comparison, we created TnT-H, a hierarchical variant of TnT requiring substantial re-architecture. As shown in Table 2, while TnT-H's overall agreement (0.60) is an improvement over flat TnT (0.59), it remains significantly lower than CLIMB's 0.73. The per-level analysis in Table 4 reveals the key insight: CLIMB achieves superior agreement at every single level of the hierarchy. This demonstrates that building a high-quality hierarchy is a complex reasoning task requiring more than repeated concept generation, but demands the specialized reflection-based framework that CLIMB provides. TnT-H's slightly higher overall utilization (61.77% vs. 54.50%) is a misleading artifact: it stems from a poorly defined upper hierarchy where only 33.33% of top-level nodes are actually used, compared to CLIMB's efficient 97.62%.

CLIMB excels at producing a taxonomy that is both comprehensive and efficient, striking a superior balance between coverage and utilization. In contrast, both baselines struggle with trading-off between these metrics. TnT achieves high label utilization only at the cost of comprehensiveness, failing to classify over 20% of jobs in the USA dataset. Conversely, ESCO provides high coverage but suffers from extremely low utilization, indicating a bloated structure. CLIMB, however, delivers strong performance on both fronts. On the USA dataset, its leaf nodes provide excellent coverage (97.34%). At a comparable size (L2), CLIMB achieves a similarly high utilization rate (82.28% vs. 82.56%) while achieving superior coverage.

CLIMB generates taxonomies whose sizes scale logically with the size and complexity of the input corpus. As shown in Table 1, the taxonomy generated by CLIMB for the large USA dataset is substantially larger than for the smaller Botswana and Palestine datasets, reflecting the greater diversity of occupations. In contrast, the TnT baseline exhibits inconsistent scaling behavior, producing a smaller taxonomy for the largest dataset (130 nodes for USA) than for the mid-sized one (200 for Botswana). This suggests that CLIMB is more robust and sensitive to the underlying occupational diversity of the corpus, whereas TnT may fail to capture the full spectrum of occupations in larger, more complex datasets.

5.2 Qualitative Analysis

Some unique structures can be found from the generated taxonomies for the USA, Palestine, and Botswana datasets, respectively. The qualitative analysis demonstrates CLIMB is indeed data-driven, and highlights CLIMB's ability to create taxonomies that reflect the unique socio-economic characteristics of each region, an advantage over any generic occupation taxonomy.

The taxonomy from the **Palestine dataset** (Figure 2(a)) reveals the prominence of humanitarian work, reflecting the region's socio-political context. The generated hierarchy dedicates one of its six top-level sectors to "Development, Humanitarian, and Community Program Management," and related jobs appear across most other branches. This cross-cutting presence allows CLIMB to capture specialized roles like Monitoring, Evaluation, Accountability, and Learning (MEAL) Specialist and

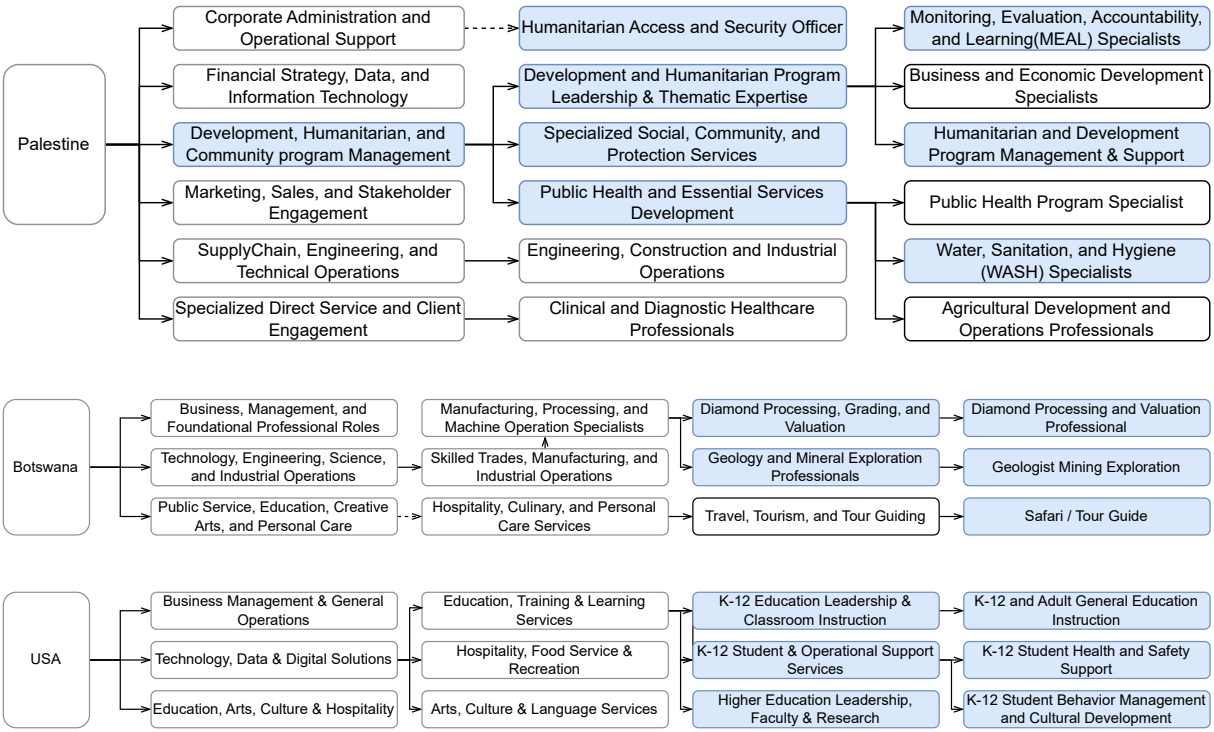


Figure 2: (a) A snippet from the Palestine taxonomy, showcasing how CLIMB identifies specialized humanitarian roles (e.g., MEAL Specialist) reflecting the local context. (b) A snippet from the Botswana taxonomy, showcasing how CLIMB reflects key economic drivers with roles in the diamond and tourism industries. (c) A snippet from the USA taxonomy, showing how CLIMB captures the specific structure of the American education system (K-12 vs. Higher Education). Dotted lines represent omitted intermediate nodes for clarity.

Water, Sanitation, and Hygiene (WASH) Specialist. The discovery of such niche, region-specific roles that would not appear in generic taxonomies is direct evidence of CLIMB's truly bottom-up, data-driven nature.

The taxonomy for the **Botswana dataset** (Figure 2(b)) mirrors the country's key economic drivers. It captures the granularity of its cornerstone diamond industry with specific roles like Diamond Processing, Grading, and Valuation. It also identifies occupations in the booming tourism sector, such as Safari Guide, which leverages the nation's natural beauty and wildlife.

For the **USA dataset** (Figure 2(c)), the CLIMB-generated taxonomy accurately models the American education system, distinctly structured into K-12 and higher education sectors. This level of regional specificity is absent in international standards like ESCO, demonstrating CLIMB's capacity to capture nuanced local labor market structures.

6 Conclusion

In this work, we introduced CLIMB, a novel framework that automatically builds hierarchical occupation taxonomies directly from raw job postings. Our approach addresses the need for up-to-date taxonomies that fit specific regional job markets. It works from the bottom up, using semantic clustering to discover occupations and a multi-agent system to build a coherent hierarchy, all without needing seed terms or manual intervention.

Our experiments on three diverse, real-world datasets show that CLIMB outperforms existing methods, producing taxonomies that are demonstrably clearer, more coherent, and achieve a better balance of coverage and efficiency. Furthermore, our qualitative analysis confirms that the resulting taxonomies capture unique, regional labor market characteristics that generic models miss.

While our implementation is tailored to job postings, we believe the core principles of the CLIMB framework are domain-agnostic. Its two-stage methodology, distilling concepts from a noisy corpus and then organizing them with a reasoning engine, could be adapted for creating taxonomies in other domains, such as scientific literature or product catalogs. This offers a promising avenue for future research into automated knowledge organization.

Limitations

Our study has several limitations that provide avenues for future work. *First*, while our evaluation includes direct validation against the expert-curated ESCO taxonomy and demonstrates CLIMB's superior fit for specific regional labor markets, a formal evaluation with domain specialists for each specific region would provide additional validation. Our methodology also relied on LLMs as proxy HR experts for data annotation and as independent annotators for evaluation. While we used a diverse panel of powerful models to mitigate risks and conducted a detailed annotator analysis (see Appendix E), their judgments may still contain biases or errors that do not perfectly reflect human expertise. Therefore, a large-scale validation study with human domain experts for both the generated training data and the final taxonomy quality is a critical next step for future work. We acknowledge the potential risk of circular validation when using LLMs for both creation and evaluation. We mitigated this through three key strategies: (1) using distinct models for distinct roles (e.g., GPT-4o-mini for training data, Gemini Pro for generation, and a completely independent panel of Gemini Flash, O4-Mini, and DeepSeek R1 for evaluation), (2) ensuring fundamentally different cognitive tasks at each stage (binary similarity judgments vs. global hierarchical reasoning vs. multi-class classification), and (3) measuring consensus across a diverse evaluation panel rather than relying on a single model, where high inter-annotator agreement indicates objective clarity that transcends individual model biases.

Second, our use of Affinity Propagation for clustering, while effective on our datasets, has quadratic time complexity that may challenge scalability on much larger corpora; future work could explore alternatives like HDBSCAN.

Third, while we created TnT-H as a hierarchical variant of TnT for a stronger comparison on the Palestine dataset, applying this computationally expensive approach to all datasets was beyond our resource constraints. Future work could extend this comparison to additional datasets.

Finally, the performance of CLIMB is inherently tied to the capabilities of the LLMs used. This includes the potential for semantic failures during hierarchy construction, where agents might create illogical groupings. Future advancements in LLM reasoning will likely improve taxonomy quality.

Acknowledgments

The research leading to these results has received funding from the Special Research Fund (BOF) of Ghent University (BOF20/IBF/117), from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, from the FWO (project no. G0F9816N, 3G042220, G073924N). Funded/Co-funded by the European Union (ERC, VIGILIA, 101142229). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. For the purpose of Open Access the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

Anthropic. 2024. Building effective agents. https://www.anthropic.com/engineering/building-effective-agents. Accessed: 2025-06-29.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*.

European Commission. 2024. Esco v1.2.0. European Skills, Competences, Qualifications and Occupations. Released 13 May 2024 by the European Commission in collaboration with Cedefop.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Michael Gunn, Dohyun Park, and Nidhish Kamath. 2024. Creating a fine grained entity type taxonomy using llms. *arXiv preprint arXiv:2402.12557*.

Jiaxin Huang, Yiqing Xie, Yu Meng, Yunyi Zhang, and Jiawei Han. 2020. Corel: Seed-guided topical taxonomy construction by concept learning and relation transferring. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1928–1936.

International Labour Organization. 2008. International Standard Classification of Occupations (ISCO-08). Resolution adopted by the International Conference

of Labour Statisticians, 2008. Available from ILO: https://www.ilo.org/public/english/bureau/stat/isco/isco08/.

Aditya Kalyanpur, Kailash Karthik Saravanakumar, Victor Barres, Jennifer Chu-Carroll, David Melville, and David Ferrucci. 2024. Llm-arc: Enhancing llms with an automated reasoning critic. *arXiv preprint arXiv:2406.17663*.

Priyanka Kargupta, Nan Zhang, Yunyi Zhang, Rui Zhang, Prasenjit Mitra, and Jiawei Han. 2025. Taxoadapt: Aligning llm-based multidimensional taxonomy construction to evolving research corpora. *arXiv preprint arXiv:2506.10737*.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. Table B interpreted in AHRQ executive summary. Ranges reproduced in AHRQ/NCBI executive summary (2012).

Thu Thi Le, Tuan-Dung Cao, Lam Xuan Pham, Trung Duc Pham, and Toan Luu. 2023. An automatic method for building a taxonomy of areas of expertise. In *ICAART*, pages 169–176.

Dongha Lee, Jiaming Shen, SeongKu Kang, Susik Yoon, Jiawei Han, and Hwanjo Yu. 2022. Taxocom: Topic taxonomy completion with hierarchical discovery of novel topic clusters. In *Proceedings of the ACM Web Conference 2022*, pages 2819–2829.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Oleksandr Marchenko and Danylo Dvoichenkov. 2024. Taxorankconstruct: A novel rank-based iterative approach to taxonomy construction with large language models. In *Proceedings of the Information Technology and Implementation (IT&I) Workshop: Intelligent Systems and Security (IT&I-WS 2024: ISS), Kyiv, Ukraine, November 20 - 21, 2024*, volume 3933 of *CEUR Workshop Proceedings*, pages 11–27. CEUR-WS.org.

Mary L. McHugh. 2012. Interrater reliability: The kappa statistic. *Biochemia Medica*.

Daniel de S Moraes, Pedro TC Santos, Polyana B da Costa, Matheus AS Pinto, Ivan de JP Pinto, Álvaro MG da Veiga, Sergio Colcher, Antonio JG Busson, Rafael H Rocha, Rennan Gaio, and 1 others. 2024. Using zero-shot prompting in the automatic creation and expansion of topic taxonomies for tagging retail banking transactions. *arXiv preprint arXiv:2401.06790*.

Cezar Sas and Andrea Capiluppi. 2024. Automatic bottom-up taxonomy construction: A software application domain study. *arXiv preprint arXiv:2409.15881*.

Chirag Shah, Ryen White, Reid Andersen, Georg Buscher, Scott Counts, Sarkar Das, Ali Montazer, Sathish Manivannan, Jennifer Neville, Nagu Rangan, and 1 others. 2023. Using large language models to generate, validate, and apply user intent taxonomies. *ACM Transactions on the Web*.

Jingbo Shang, Xinyang Zhang, Liyuan Liu, Sha Li, and Jiawei Han. 2020. Nettaxo: Automated topic taxonomy construction from text-rich network. In *Proceedings of the web conference 2020*, pages 1908–1919.

Jiaming Shen, Zeqiu Wu, Dongming Lei, Chao Zhang, Xiang Ren, Michelle T Vanni, Brian M Sadler, and Jiawei Han. 2018. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2180–2189.

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W White, Longqi Yang, and 1 others. 2024. Tnt-llm: Text mining at scale with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5836–5847.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Yurun Yuan and Tengyang Xie. 2025. Reinforce LLM reasoning through multi-agent reflection. In *Forty-second International Conference on Machine Learning*.

Qingkai Zeng, Yuyang Bai, Zhaoxuan Tan, Shangbin Feng, Zhenwen Liang, Zhihan Zhang, and Meng Jiang. 2024. Chain-of-layer: Iteratively prompting large language models for taxonomy induction from limited examples. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 3093–3102.

Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. 2018. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2701–2709.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Job posting datasets construction

The three real-world datasets used in our experiments were constructed as follows:

- **Palestine Dataset**: Contains 2,701 job postings from `https://www.jobs.ps/`, a major job portal in Palestine. The data spans from September 2024 to February 2025.
- **Botswana Dataset**: Contains 4,854 job postings from `https://jobsbotswana.info/`, a leading job board in Botswana. The data covers the period from August 2023 to June 2025.
- **USA Dataset**: Contains 11,285 job postings from `https://www.indeed.com/`, with the location restricted to the United States. Data was collected between October 2023 and November 2023.

B Dataset Characteristics and Challenges

This section provides concrete examples from our datasets to illustrate the key challenges in the job-postings domain that CLIMB is designed to overcome. The following examples showcase the complexities of real-world job data that necessitate our proposed methodology.

High Noise-to-Signal Ratio Job postings are frequently cluttered with non-essential information, such as company boilerplate, benefits, and application instructions, which can obscure the core occupational details. As shown in Table 5, the description for a "Bus Driver" position is dominated by administrative details rather than the actual duties and qualifications. This high noise-to-signal ratio makes it difficult to extract meaningful information and justifies our initial distillation step, which filters out irrelevant text to focus on the essential aspects of the job.

Synonymy and Ambiguous Titles The same occupation is often described using a variety of synonymous or ambiguous titles. For instance, as illustrated in Figure 3, roles like "Administrative Assistant," "Administrative Support," and "Administrative Assistant III" may refer to very similar jobs, yet their titles differ. This ambiguity makes it challenging to group similar occupations based on job titles alone and motivates our use of semantic clustering, which relies on the underlying meaning of job descriptions rather than superficial title variations.

Niche, Region-Specific Roles Labor markets contain many specialized roles that are specific to a particular region or industry and are often absent from standardized, top-down taxonomies. Figure 4 highlights one such example, the "Water, Sanitation and Hygiene (WASH) Officer," a role prevalent in the international development sector. The discovery of such niche roles underscores the value of CLIMB's bottom-up approach, which can adaptively identify and categorize emerging or region-specific occupations from the data itself.

Multilingual Content Job posting datasets can contain a mix of languages, adding another layer of complexity. The Palestine dataset, for example, includes job postings in both English and Arabic, as shown in Figure 5. Processing such multilingual content requires models that can understand and compare job descriptions across different languages, which guided our choice of multilingual embedding and language models throughout the CLIMB pipeline.

C Implementation Details

This section provides a detailed breakdown of each stage in the CLIMB pipeline, complementing the descriptions in Section 3.

C.1 Job Posting Distillation & Embedding

The goal of this initial stage is to efficiently extract core occupational information from job postings, which are often verbose and contain irrelevant text (e.g., company boilerplate, application instructions). Applying a large language model (LLM) to summarize every posting in a large corpus would be prohibitively slow and expensive. To address this scalably, we adopt a two-step approach: first, we use a cost-effective LLM to generate a high-quality labeled dataset, and second, we train a lightweight yet effective classifier to perform the distillation on the entire corpus. This process ensures that the subsequent clustering stage operates on clean, relevant data.

- **Data Preparation**: Job descriptions first undergo basic cleaning to remove HTML tags and excess whitespace. They are then segmented into text chunks by paragraphs (split by new lines)
- **LLM-based Labeling for Training Data**: To create training data for the distillation classifier, we use `gpt-4o-mini` to annotate a sample of the text chunks. Each chunk is la-

Table 5: An Example of Highly Noisy Job Posting.

Job Title	Job Posting
Bus Driver	Benefits: One year signing bonus of $1000, health insurance, NYS pension, paid holidays, paid time off, and flexible work hours. Training Provided (CDL license with P&S Endorsement) Job Attachment Itinerant Location District wide Salary: From 22.50 Salary: To 22.50 Start Date ASAP Additional Job Information: Salary listed is the minimum starting rate for the position. Compensation may be commensurate with experience. Must posses or be willing to obtain a valid Commercial Driver's License - Class B with Passenger (P) and (S) endorsements, airbrakes by the State of New York. Contact Person's Name: Finune O. Shaibi Contact Person's Title: Director of Human Resources Contact Person's Phone: 716-250-1413 Contact Person's Email: jgalbo@sweethomeschools.org

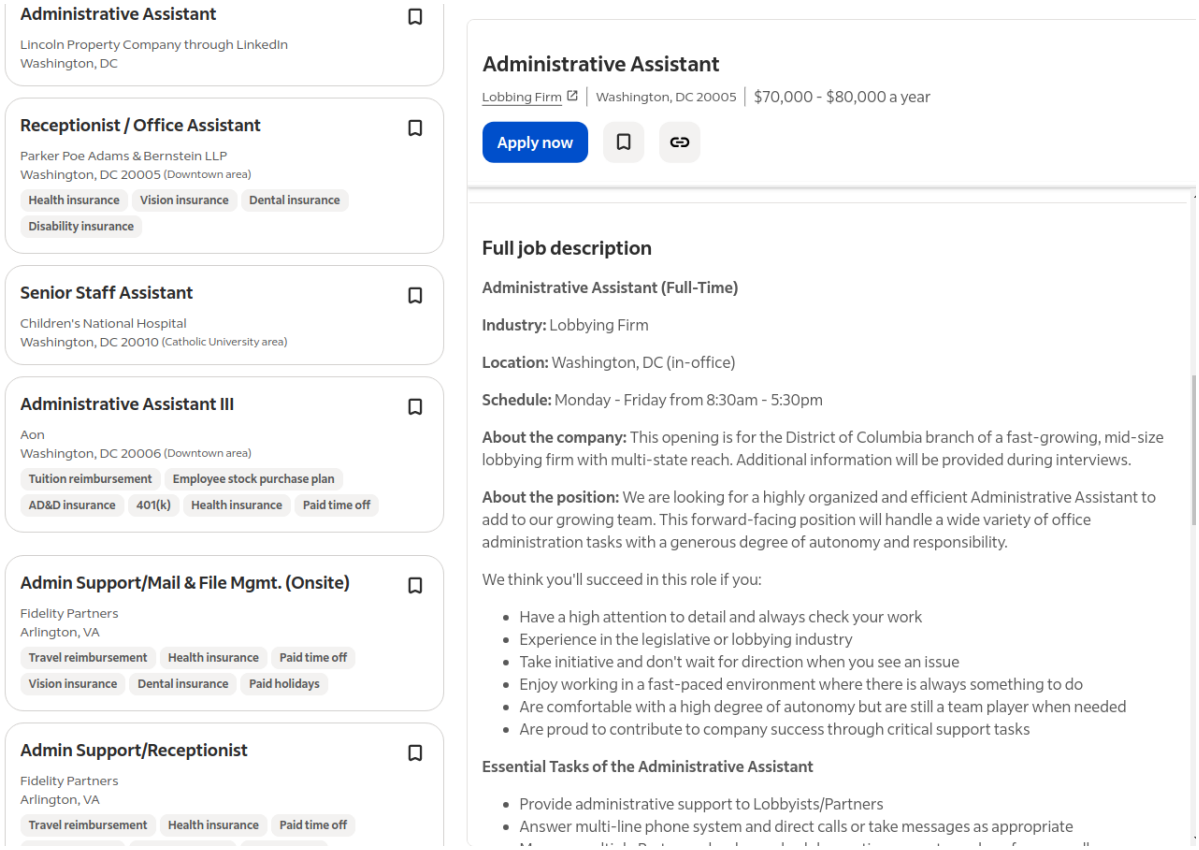
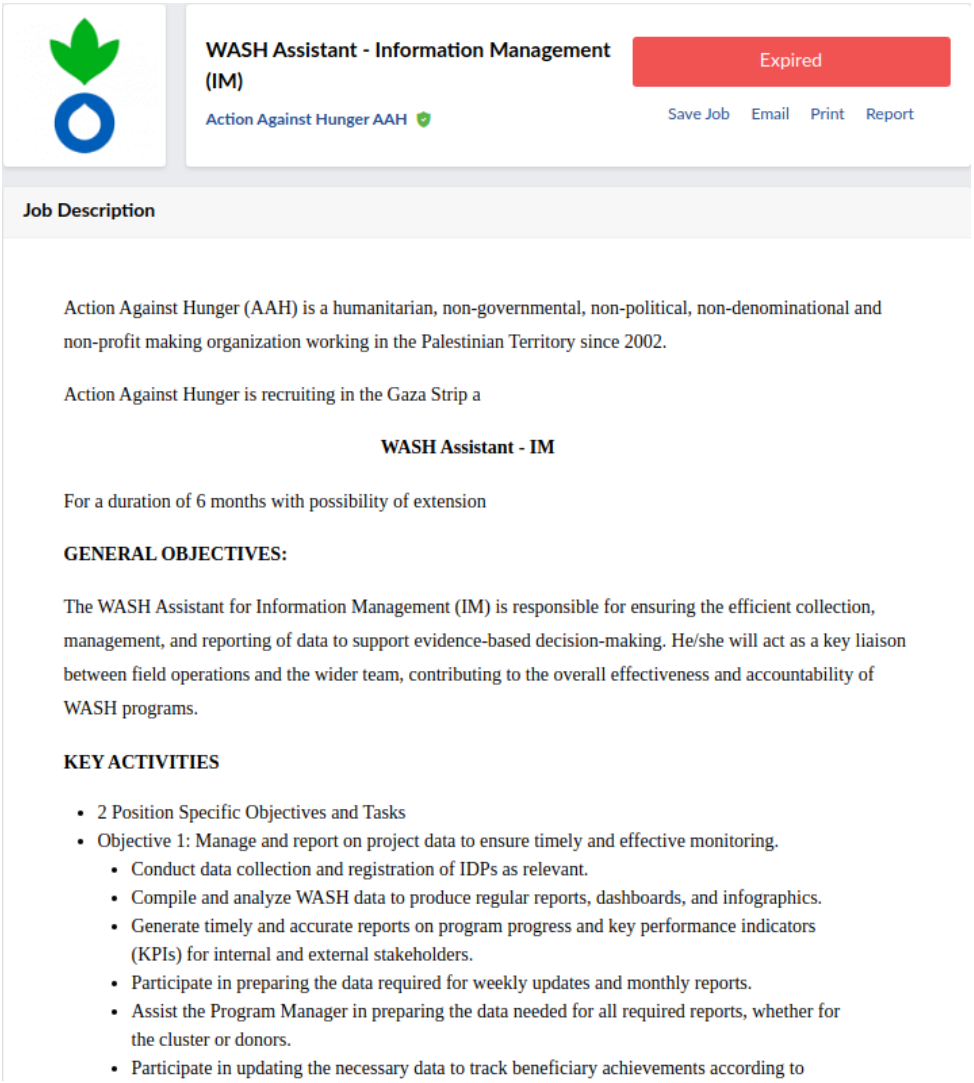


Figure 3: Synonymy and Ambiguous Titles for Administrative Support Occupations.

beled as either relevant or irrelevant to defining the core occupation. The prompt guided the LLM to identify text segments directly describing job duties, responsibilities, and required qualifications, while ignoring other content. The full prompt details are available at https://github.com/aida-ugent/CLIMB/src/annotate_posting_segments.py.

- **Distillation Classifier Training**: A lightweight binary classifier is trained on the LLM-annotated data to automate the relevance-filtering process for the entire corpus.
  - **Data Split**: The LLM-labeled dataset of text chunks was split into training (90%) and testing (10%) sets. The training set was further divided into training (90%) and validation (10%) subsets from training the classifier.
  - **Model Architecture**: The classifier is an `XLMRobertaForSequenceClassification` model initialized with `BAAI/bge-m3` weights,



WASH Assistant - Information Management (IM)

Action Against Hunger AAH

Expired

Save Job Email Print Report

Job Description

Action Against Hunger (AAH) is a humanitarian, non-governmental, non-political, non-denominational and non-profit making organization working in the Palestinian Territory since 2002.

Action Against Hunger is recruiting in the Gaza Strip a

WASH Assistant - IM

For a duration of 6 months with possibility of extension

GENERAL OBJECTIVES:

The WASH Assistant for Information Management (IM) is responsible for ensuring the efficient collection, management, and reporting of data to support evidence-based decision-making. He/she will act as a key liaison between field operations and the wider team, contributing to the overall effectiveness and accountability of WASH programs.

KEY ACTIVITIES

- 2 Position Specific Objectives and Tasks
- Objective 1: Manage and report on project data to ensure timely and effective monitoring.
  - Conduct data collection and registration of IDPs as relevant.
  - Compile and analyze WASH data to produce regular reports, dashboards, and infographics.
  - Generate timely and accurate reports on program progress and key performance indicators (KPIs) for internal and external stakeholders.
  - Participate in preparing the data required for weekly updates and monthly reports.
  - Assist the Program Manager in preparing the data needed for all required reports, whether for the cluster or donors.
  - Participate in updating the necessary data to track beneficiary achievements according to

Figure 4: An Example of Region-Specific Roles: Water, Sanitation and Hygiene (WASH) Officer.

featuring a classification head to output the binary prediction.

  - **Training and Performance**: The model was trained for 1 epoch with a learning rate of 2e-5 and 100 warmup steps, using the AdamW optimizer. On the USA dataset, it achieved a test accuracy of 93.58%, precision of 93.18%, recall of 95.98%, and an F1-score of 94.56%, demonstrating its effectiveness in identifying relevant content.

- **Distilled Description Generation**: After training, the classifier is applied to all text chunks in the corpus. For each job posting, the chunks classified as relevant are concatenated to form a "distilled" description.

- **Fallback Mechanism**: For robustness, if the distillation process results in no relevant text for a given posting (an infrequent event that occurred for only 9 jobs in the USA dataset), the full, pre-processed job description is used as a fallback to ensure no data is lost.

- **Embedding for Clustering**: Finally, the resulting distilled (or full) descriptions are embedded using the `Qwen3-Embedding-8B` model. This produces the final vector representations that serve as the input for the Semantic Clustering stage.

C.2 Semantic Clustering

This stage groups job postings into fine-grained clusters representing distinct occupations. The core idea is to train a custom similarity model that learns to mimic a human's nuanced judgment of what constitutes the "same occupation," rather than relying on generic cosine similarity, which often falls short.

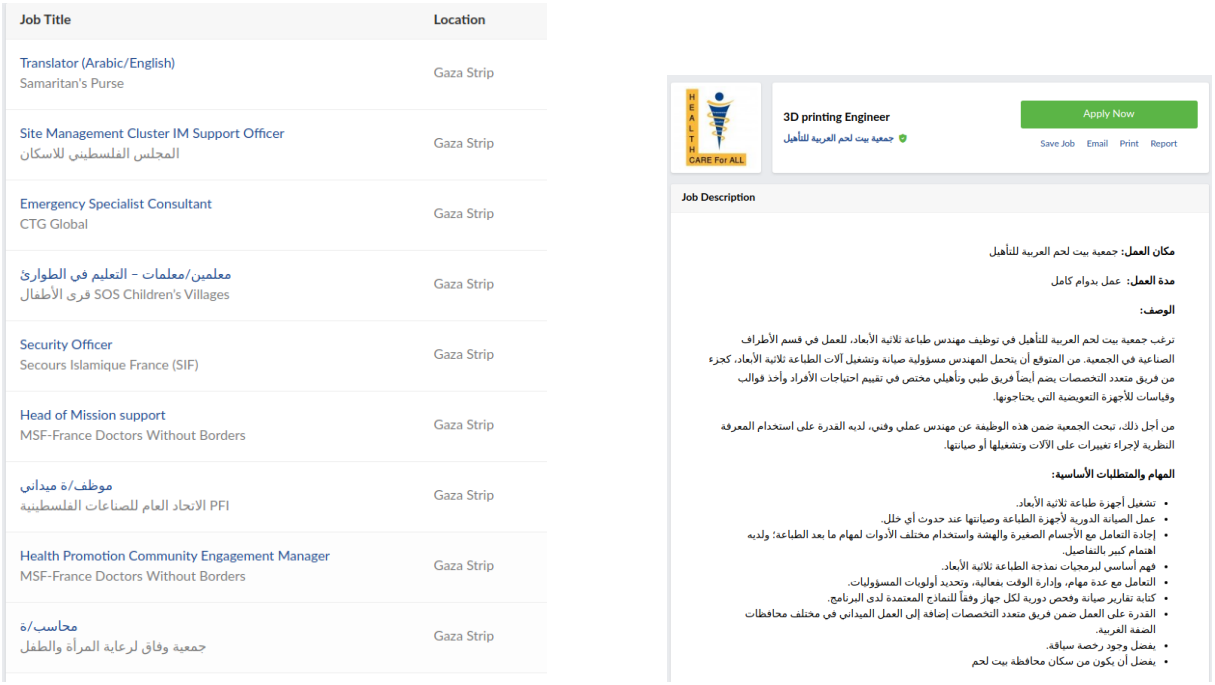


Figure 5: Multilingual Content in the Palestine Dataset.

This learned similarity is then used to drive a clustering algorithm.

- **Contrastive Data Sampling**: To train the similarity model, we first create a challenging set of job pairs. For each job in a dataset, we compute its cosine similarity with all other jobs using their embeddings. Based on this, we sample pairs to create a balanced mix of examples:
  - **Likely Easy Positives**: Two strongly similar jobs (from nearest neighbors ranked 1-20).
  - **Likely Ambiguous Hard-Negatives or Hard-Positives**: One job with moderate similarity (from neighbors ranked 21-100).
  - **Likely Easy Negatives**: One job with low similarity (from the rest of the corpus).

  This strategy deliberately includes ambiguous (hard-negative or hard-positive) pairs to teach the model a more nuanced understanding. For the USA dataset, this process resulted in approximately 40,000 pairs for training.

- **LLM as HR Expert**: The sampled pairs are then annotated by `gpt-4o-mini`, which acts as a proxy for an HR expert. It labels each pair as "same occupation" or "different occupation" from the practical perspective of a job seeker. The full prompt is available at `https://github.com/aida-ugent/CLIMB/src/same_occupation_job_pair_sampling.py`.

- **Similarity Classifier Training**: An XGBoost classifier is trained on the LLM-annotated pairs to predict the probability that two jobs belong to the same occupation.
  - **Features**: For a pair of job embeddings, e_a and e_b, the input feature vector is the concatenation of $(e_a, e_b, e_a - e_b, e_a \odot e_b)$, where $\odot$ denotes the Hadamard (element-wise) product. This feature engineering captures rich interaction information between the job descriptions.
  - **Hyperparameters**: Key hyperparameters for the XGBoost model include a learning rate (`eta`) of 0.1, a `max_depth` of 8, and a `subsample` ratio of 0.8. To account for class imbalance in the training data, `scale_pos_weight` was set to approximately 1.67. The model was trained with a maximum of 2000 boosting rounds, using early stopping with a patience of 50 rounds.
  - **Training and Model Selection**: The dataset of labeled job pairs was split into training (90%) and testing (10%) sets. This setup was used to evaluate different embedding models for representing the jobs, including BGE-m3, Qwen3-Embedding-0.6B, Qwen3-Embedding-4B, and Qwen3-Embedding-8B. The Qwen3-Embedding-8B model was ultimately selected as it yielded the best classification performance on the test set.

- **Similarity Matrix Construction**: After training, the XGBoost classifier is used to compute a pairwise similarity score for all jobs in the training set. To ensure the similarity matrix is symmetric (i.e., $\text{sim}(A, B) = \text{sim}(B, A)$), we define the final similarity between two jobs as the average of the two predictions: $(\text{score}(A, B) + \text{score}(B, A))/2$.

- **Clustering Algorithm Selection**: We explored several clustering algorithms, including Agglomerative Clustering, a custom greedy approach, and Affinity Propagation. Affinity Propagation was selected as it consistently yielded the best performance, as measured by the silhouette score on the generated clusters.

- **Final Clustering**: The Affinity Propagation algorithm is used to cluster the job postings based on the symmetrized similarity matrix. We used the default damping factor of 0.5 and set the preference hyperparameter to the median of the input similarities, which allows the algorithm to determine the number of clusters automatically.

C.3 Leaf Node Generation

This stage transforms the raw job posting clusters from the previous step into canonical, well-defined leaf nodes, each with a clear title and description, which serve as the foundation of the taxonomy. This involves two main sub-stages: abstracting a canonical occupation from each cluster's raw text and then refining the full set of generated occupations to ensure consistency and remove redundancy.

- **LLM-based Abstraction**: The goal of this step is to synthesize the content of each job cluster into a single, representative occupation.
  - **Model and Input**: For each cluster, we provide a sample of its raw job descriptions to the `gemini-2.5-flash-preview-05-20` model. Using a sample is necessary to manage the context window limits for clusters containing a large number of postings.
  - **Prompting**: The LLM is prompted to act as an HR expert and generate a concise, generic title and a comprehensive description that canonically represents the occupation for the given job descriptions. The full prompt is available at https://github.com/aida-ugent/CLIMB/src/prompts.py.

- **Normalization and Deduplication**: The raw LLM-generated titles can be inconsistent or semantically redundant (e.g., "Software Engineer" vs. "Software Developer"). To create a clean set of leaf nodes, we perform a three-step refinement process:
  - **Filtering**: Nodes with conjoined titles (e.g., "Accountant + Bookkeeper") are programmatically removed by detecting the "+" separator. These titles were intentionally generated by the LLM for clusters spanning multiple distinct occupations but are filtered out as they violate the principle of a node representing a single, distinct concept.
  - **Cleaning**: Standard text normalization is applied to all titles for consistency. This includes converting text to lowercase, removing punctuation and stopwords, and performing lemmatization.
  - **Deduplication**: To merge semantically equivalent nodes, we cluster them based on their meaning.
    i. **Embedding**: The cleaned title and the full description of each node are embedded using the `Qwen3-Embedding-8B` model.
    ii. **Feature Representation**: A final vector for each node is created by concatenating its title embedding and description embedding. To emphasize the title's importance while retaining contextual information from the description, the title embedding is weighted at 80% and the description embedding at 20%.
    iii. **Clustering**: Affinity Propagation is applied to the cosine similarity matrix of these combined embeddings. The algorithm was configured with a damping factor of 0.7, and the preference was set to 0.95.
    iv. **Canonicalization**: For each resulting group of semantically similar nodes, the "exemplar" identified by Affinity Propagation is chosen as the single canonical representative, and the other nodes in the group are merged. This step ensures each distinct occupation is represented by only one node in the final set.

C.4 Hierarchical Taxonomy Construction

This final stage constructs the taxonomy by building a hierarchy upwards from the canonical leaf nodes generated in the previous step. Constructing a deep and logically coherent hierarchy is a complex reasoning task. Our early explorations revealed that single-pass LLM approaches are prone to critical errors, such as inconsistent groupings or flawed parent-child relationships, which corrupt the entire structure. To address this, we designed a deliberate, level-by-level process using a reflection-based multi-agent framework. This ensures logical coherence at each step of the construction. The process is governed by two agents, a "Generator" and an "Evaluator," operating in an iterative cycle.

- **Framework and Input**: The process begins with the set of canonical leaf nodes (Level 0), which are formatted as a JSON list of objects, each with a "title" and "description". The multi-agent framework iteratively processes this list to build the hierarchy one level at a time.
- **Generator Agent**:
  - **Model**: `gemini-2.5-pro-preview-05-06`. Specific generation hyperparameters (e.g., temperature) are detailed in the codebase, available at https://github.com/aida-ugent/CLIMB/src/tree_multiagent.py.
  - **Task**: At each level k, the Generator's task is to take the set of nodes from that level and propose a set of parent concepts for the next level, $k+1$. This involves performing specific-to-general abstraction to group the input concepts into broader parent categories.
  - **Prompting and Output**: The agent uses a detailed prompt instructing it to return a JSON object containing the list of new parent nodes. For each parent, it must provide a concise "title", a comprehensive "description", and a list of the child nodes from level k that it subsumes. The prompt explicitly requires strict adherence to this JSON schema to ensure the output is machine-readable. It also includes a mechanism for receiving feedback from the Evaluator, allowing it to correct errors from previous attempts. The complete prompt templates are available at https://github.com/aida-ugent/CLIMB/src/tree_multiagent.py.
- **Evaluator Agent**: Unlike the Generator, the Evaluator is a deterministic, rule-based agent. It programmatically scrutinizes the Generator's output to ensure its logical coherence before it is accepted as a valid level in the hierarchy. It performs several critical checks:
  - **Completeness**: All child nodes from level k must be mapped to a parent in level $k+1$.
  - **Exclusivity**: No child node can be assigned to more than one parent, enforcing a strict tree structure.
  - **Validity**: There are no "hallucinated" mappings to child nodes that did not exist in the input.
  - **Constraints**: The number of generated parent nodes must be within a pre-defined range (see below).
- **Iterative Generate-Evaluate Cycle**: The construction proceeds in a loop. If the Evaluator finds flaws in the Generator's output, its feedback is incorporated into the prompt for the next generation attempt. This cycle repeats until the output is fully validated. Once a level is validated, its newly generated parent nodes become the input for constructing the subsequent level. This process continues until the taxonomy converges to a small set of top-level categories, with the final hierarchy stored in a JSON file.
- **Grouping and Termination Constraints**: To guide the hierarchical construction, the process is bounded by the following constraints:
  - **Dynamic Grouping**: The number of parent nodes generated at each level is dynamic to adapt to the data's complexity.
  - **Termination**: The entire process terminates when either of two conditions is met: (1) the number of generated parent nodes at a new level is less than a threshold of 10, or (2) the number of nodes to be clustered in the current level is less than or equal to 1. This prevents the creation of overly granular or trivial top-level categories.

C.5 Evaluation

To evaluate the quality of the generated taxonomies scalably and cost-effectively, we used a panel of three distinct LLMs as independent annotators. Each annotator was tasked with labeling every job posting in the test set according to a given taxonomy. This process is detailed below:

- **Annotator Panel**: The panel consisted of three models from different providers to ensure a diversity of perspectives: `o4-mini`, `gemini-2.5-flash-preview-04-17`, and `deepseek/deepseek-r1-0528`.

- **Annotation Task and Prompting**: For each job posting, an LLM was provided with the full job description and a string representation of one of the taxonomies (CLIMB, TnT, or ESCO). The prompt instructed the LLM to act as a job classification expert and adhere to several rules: (1) assign the most granular label possible, (2) use parent nodes for vague postings, (3) use multiple labels only for jobs that genuinely span distinct roles, and (4) use an "Other" category if no suitable label exists. The prompt also strictly enforced a JSON output format for programmatic parsing. The full prompt templates are available in the codebase at https://github.com/aida-ugent/CLIMB/src/taxonomy_evaluation.py.

- **Execution and Parsing**: API calls were made with a temperature of 0.0 to ensure deterministic outputs. The final assigned labels for each job were then extracted from the returned JSON object to be used for calculating the evaluation metrics described in the main text.

D Taxonomy Trees

The taxonomies generated by CLIMB for each dataset have the following hierarchical structures:

- **Palestine Dataset**: A 4-level taxonomy. The number of nodes per level, from the most general (top) to the most specific (bottom), is 6, 14, 32, and 91. The interactive visualization of the taxonomy is online https://github.com/aida-ugent/CLIMB/file/demo/palestine.html.

- **Botswana Dataset**: A 5-level taxonomy. The number of nodes per level, from most general to most specific, is 3, 10, 29, 99, and 277. The interactive visualization of the taxonomy is online https://github.com/aida-ugent/CLIMB/file/demo/botswana.html.

- **USA Dataset**: A 6-level taxonomy. The number of nodes per level, from most general to most specific, is 8, 23, 65, 190, 341, and 671. The interactive visualization of the taxonomy is online https://github.com/aida-ugent/CLIMB/file/demo/usa.html.

E LLM Annotator Analysis

This section provides a detailed characterization of our three-model evaluation panel used for taxonomy annotation. We analyze (1) quantitative metrics of annotator bias including positional bias and label diversity, (2) pairwise inter-model agreement to assess panel consistency, and (3) the generation hyperparameters employed.

E.1 Evaluation Panel Configuration

Our annotation pipeline employed a three-model ensemble consisting of:

- **DeepSeek-R1-0528**: A reasoning-focused model with chain-of-thought capabilities,

- **Gemini-2.5-Flash-Preview-04-17**: Google's latest multimodal model,

- **O4-Mini**: A compact yet capable model optimized for classification tasks.

All models were configured with **temperature = 0.0** to ensure deterministic and reproducible annotations. Each model independently annotated job descriptions across three taxonomies (ESCO, FLAT, TREE) on three geographic datasets (USA, Palestine, Botswana), totaling approximately 1,200 annotations per model per taxonomy.

E.2 Annotator Bias Analysis

We assess two critical dimensions of annotator quality: *label diversity* (measuring whether models exploit the full taxonomy) and *positional bias* (measuring preference for labels based on their position in the taxonomy list).

E.2.1 Bias Metrics: USA Dataset

Table 6 presents comprehensive bias metrics for the USA dataset, our largest evaluation set with ~1,160 annotations per model.

Key findings: All three models demonstrate high label diversity, with normalized entropy consistently above 0.85 across all taxonomies. Notably, on the fine-grained ESCO taxonomy, models achieved entropy scores above 0.90, indicating near-uniform exploitation of the label space. Positional bias remains well below the random baseline of 0.25 for all models, with most values between 0.11–0.30, suggesting minimal preference for early-listed candidates.

Table 6: Annotator bias metrics on USA dataset across three taxonomies. Normalized entropy quantifies label diversity (1.0 = perfectly uniform). Primacy bias measures preference for early-listed labels (0.25 = random baseline). Combined bias score synthesizes both metrics (higher = less biased).

Taxonomy	Model	Annot. Count	Unique Labels	Norm. Entropy	Primacy Bias	Position Entropy	Combined Bias Score
ESCO	DeepSeek-R1	1160	404	0.914	0.146	0.915	0.781
	Gemini-2.5-Flash	1084	397	0.928	0.304	0.928	0.645
	O4-Mini	1153	337	0.906	0.190	0.906	0.734
FLAT	DeepSeek-R1	942	112	0.879	0.165	0.879	0.734
	Gemini-2.5-Flash	865	102	0.867	0.109	0.867	0.773
	O4-Mini	971	108	0.852	0.160	0.852	0.716
TREE	DeepSeek-R1	1165	374	0.918	0.350	0.918	0.596
	Gemini-2.5-Flash	1089	448	0.950	0.210	0.950	0.751
	O4-Mini	1148	425	0.935	0.266	0.935	0.687

E.2.2 Label Distribution Characteristics

Table 7 shows the top-5 most frequently assigned labels for each model on the USA dataset, revealing annotation patterns.

Even the most frequently assigned label ("Administrative assistant") accounts for only 3.5–5.5% of annotations, confirming that no single label dominates model predictions.

E.3 Inter-Model Agreement Analysis

We assess pairwise agreement between models using four complementary metrics:

- **Exact Agreement**: Fraction of items where both models assign identical label sets
- **Loosened Agreement**: Fraction of items with any label overlap (accommodates multi-label partial matches)
- **Jaccard Similarity**: Average Jaccard index across all item pairs
- **Partial Overlap Rate**: Fraction of items with non-empty label intersection

E.3.1 Cross-Dataset Agreement Summary

Table 8 presents agreement metrics averaged across all three datasets (USA, Palestine, Botswana) and all three taxonomies.

Key findings: The Gemini–O4-Mini pair exhibits the strongest agreement (loosened agreement = 0.661), while DeepSeek–Gemini shows the lowest (0.583). All pairs demonstrate moderate to substantial agreement, with exact match rates between 56–64%. The modest gap between exact and loosened agreement (average improvement of 2.2 percentage points) indicates that most agreements are exact matches rather than partial overlaps, reflecting consistent classification behavior.

E.3.2 Taxonomy-Specific Agreement Patterns

Table 9 breaks down agreement by taxonomy across all datasets.

Key findings: Agreement varies systematically with taxonomy structure. The FLAT taxonomy achieves the highest agreement (DeepSeek–O4-Mini: 0.685 loosened), while ESCO shows the lowest (DeepSeek–Gemini: 0.489), consistent with the increased difficulty of achieving consensus in larger label spaces. The hierarchical TREE taxonomy shows intermediate agreement (0.623–0.741), with Gemini–O4-Mini achieving particularly strong consensus (0.741).

E.3.3 Dataset-Specific Agreement: USA

Table 10 presents detailed agreement metrics for the USA dataset, our primary evaluation benchmark.

On the FLAT taxonomy, Gemini–O4-Mini achieve 78.2% loosened agreement, suggesting strong consensus for this compact label space. Even on the challenging ESCO taxonomy, models reach 46–56% exact agreement, well above chance (~0.25% random baseline for 400 labels).

E.3.4 Jensen-Shannon Divergence Analysis

Table 11 presents Jensen-Shannon divergence (JSD) between model label distributions, measuring distributional similarity.

Key findings: JSD values range from 0.171 (FLAT) to 0.429 (ESCO), indicating moderate distributional alignment. The FLAT taxonomy shows remarkably low divergence (0.17–0.21), while ESCO shows higher values (0.37–0.43), consistent with the greater labeling ambiguity in fine-grained

Table 7: Top-5 most frequently assigned labels by each model (USA dataset, ESCO taxonomy). Even the most common labels represent <6% of annotations, demonstrating high diversity.

Model	Rank	Label	Frequency (%)
DeepSeek-R1	1	Administrative assistant	5.5%
	2	Security guard	2.1%
	3	Policy officer	1.9%
	4	Human resources officer	1.5%
	5	Department manager	1.4%
Gemini-2.5-Flash	1	Administrative assistant	3.5%
	2	Education managers	2.4%
	3	Executive assistant	2.1%
	4	Security guard	1.9%
	5	Medical administrative assistant	1.3%
O4-Mini	1	Administrative assistant	5.4%
	2	Office clerk	3.1%
	3	Security guard	2.0%
	4	Sales assistant	1.8%
	5	Education managers	1.7%

Table 8: Overall pairwise inter-model agreement averaged across three taxonomies (ESCO, FLAT, TREE) and three datasets (USA, Palestine, Botswana). Mean ± standard deviation reported. Higher values indicate stronger agreement.

Model Pair	Exact Agree.	Loosened Agree.	Jaccard Sim.	Partial Overlap	Tax.	Items
Gemini ↔ O4-Mini	0.641 ± 0.055	$\mathbf{0.661 \pm 0.056}$	0.651 ± 0.056	0.661 ± 0.056	3	4,976
DeepSeek ↔ O4-Mini	0.606 ± 0.061	0.626 ± 0.066	0.616 ± 0.063	0.626 ± 0.066	3	5,313
DeepSeek ↔ Gemini	0.557 ± 0.060	0.583 ± 0.067	0.570 ± 0.063	0.583 ± 0.067	3	4,946

Table 9: Inter-model agreement by taxonomy, averaged across three geographic datasets. FLAT taxonomy (112 labels) shows highest agreement; ESCO taxonomy (404 labels) shows lowest, as expected given the larger label space.

Taxonomy	Model Pair	Datasets	Exact	Loosened	Jaccard	Partial	Items
ESCO	DeepSeek ↔ Gemini	3	0.472	0.489	0.480	0.489	1,787
	DeepSeek ↔ O4-Mini	3	0.519	0.533	0.526	0.533	1,902
	Gemini ↔ O4-Mini	3	0.597	0.616	0.606	0.616	1,782
FLAT	DeepSeek ↔ Gemini	3	0.593	0.637	0.614	0.637	1,344
	DeepSeek ↔ O4-Mini	3	0.646	0.685	0.665	0.685	1,521
	Gemini ↔ O4-Mini	3	0.606	0.627	0.617	0.627	1,394
TREE	DeepSeek ↔ Gemini	3	0.607	0.623	0.615	0.623	1,815
	DeepSeek ↔ O4-Mini	3	0.653	0.660	0.656	0.660	1,890
	Gemini ↔ O4-Mini	3	0.719	0.741	0.729	0.741	1,800

Table 10: Pairwise inter-model agreement on USA dataset across three taxonomies. Common items indicates the number of job descriptions annotated by both models in each pair.

Taxonomy	Model Pair	Common Items	Exact Agree.	Loosened Agree.	Jaccard Sim.	Partial Overlap
ESCO	DeepSeek ↔ Gemini	1,081	0.449	0.463	0.456	0.463
	DeepSeek ↔ O4-Mini	1,150	0.479	0.483	0.481	0.483
	Gemini ↔ O4-Mini	1,075	0.546	0.558	0.552	0.558
FLAT	DeepSeek ↔ Gemini	796	0.690	0.724	0.706	0.724
	DeepSeek ↔ O4-Mini	887	0.692	0.720	0.706	0.720
	Gemini ↔ O4-Mini	830	0.760	0.782	0.771	0.782
TREE	DeepSeek ↔ Gemini	1,089	0.495	0.511	0.503	0.511
	DeepSeek ↔ O4-Mini	1,148	0.582	0.582	0.582	0.582
	Gemini ↔ O4-Mini	1,076	0.644	0.664	0.654	0.664

Table 11: Jensen-Shannon divergence (JSD) between model label distributions on USA dataset. JSD $\in [0, 1]$: 0 = identical distributions, 1 = completely different. Lower values indicate more similar annotation patterns.

Taxonomy	Model Pair	JSD
ESCO	Gemini ↔ O4-Mini	0.371
	DeepSeek ↔ O4-Mini	0.397
	DeepSeek ↔ Gemini	0.429
FLAT	DeepSeek ↔ O4-Mini	0.171
	Gemini ↔ O4-Mini	0.195
	DeepSeek ↔ Gemini	0.207
TREE	Gemini ↔ O4-Mini	0.324
	DeepSeek ↔ O4-Mini	0.352
	DeepSeek ↔ Gemini	0.402

taxonomies. Across all taxonomies, Gemini and O4-Mini exhibit the lowest divergence, suggesting they employ similar annotation strategies.

E.4 Summary and Implications

Our analysis demonstrates that all three models serve as high-quality annotators with:

1. **High label diversity**: Normalized entropy $>$ 0.85 across all configurations, with models utilizing 100–450 unique labels per taxonomy
2. **Low positional bias**: Primacy bias consistently below the 0.25 random baseline, indicating position-invariant evaluation
3. **Moderate to substantial inter-model agreement**: Pairwise agreement ranging from 0.58–0.74 (averaged across taxonomies), with particularly strong consensus on coarse-grained taxonomies
4. **Consistent distributional patterns**: Jensen-Shannon divergences of 0.17–0.43, demonstrating convergent annotation strategies

These metrics collectively validate our three-model ensemble as a robust evaluation panel, exhibiting both individual quality (low bias, high diversity) and collective consistency (strong pairwise agreement). The systematic variation in agreement across taxonomies (FLAT $>$ TREE $>$ ESCO) aligns with theoretical expectations: coarser taxonomies admit less ambiguity and thus higher consensus.

E.5 Generation Hyperparameters

All annotations were generated using deterministic sampling with **temperature = 0.0** to ensure reproducibility. No top-p or top-k filtering was applied. Each model processed identical prompts containing the job description, taxonomy definitions, and standardized instructions.