

Unified Work Embeddings: Contrastive Learning of a Bidirectional Multi-task Ranker

Matthias De Lange, Warre Veys, Jens-Joris Decorte, Jeroen Van Hautte

TechWolf, Ghent, Belgium
research@techwolf.ai

Abstract

Applications in labor market intelligence demand specialized NLP systems for a wide range of tasks, characterized by extreme multi-label target spaces, strict latency constraints, and multiple text modalities such as skills and job titles. These constraints have led to isolated, task-specific developments in the field, with models and benchmarks focused on single prediction tasks. Exploiting the shared structure of work-related data, we propose a unifying framework, combining a wide range of tasks in a multi-task ranking benchmark, and a flexible architecture tackling text-driven work tasks with a single model. The benchmark, WorkBench, is the first unified evaluation suite spanning six work-related tasks formulated explicitly as ranking problems, curated from real-world ontologies and human-annotated resources. WorkBench enables cross-task analysis, where we find significant positive cross-task transfer. This insight leads to Unified Work Embeddings (UWE), a task-agnostic bi-encoder that exploits our training-data structure with a many-to-many InfoNCE objective, and leverages token-level embeddings with task-agnostic soft late interaction. UWE demonstrates zero-shot ranking performance on unseen target spaces in the work domain, and enables low-latency inference with two orders of magnitude fewer parameters than best-performing generalist models that are not tailored to the work domain (Qwen3-8B), with +4.4 MAP improvement.

1 Introduction

Applications in labor market intelligence (LMI) demand specialized NLP systems capable of scaling across diverse tasks and extreme multi-label target spaces, while meeting strict latency constraints for processing unstructured data from heterogeneous sources. These requirements have driven isolated, task-specific developments in the field, with models and benchmarks targeting individual prediction tasks such as skill extraction

or job normalization (Zhang et al., 2022; Decorte et al., 2021, 2025b). This fragmentation poses both practical and scientific limitations, accumulating inference and maintenance costs over tasks, while precluding potential cross-task knowledge transfer and holistic progress measurement.

Concurrently, general-purpose text embedding models have advanced rapidly, demonstrating with comprehensive benchmarks that a single model can be effective across many standard NLP tasks (Thakur et al., 2021; Muennighoff et al., 2023). However, both evaluation and models do not account for the specifics of the work domain, including grounding in domain-specific ontologies, extreme label cardinality, and high-speed inference. This raises a key question: can a single, efficient embedding model replace multiple task-specific systems for real-world LMI applications?

To systematically investigate this question, we introduce WorkBench, a unified evaluation suite that formalizes six real-world job and skill understanding tasks as ranking problems with standardized metrics. Our analysis on WorkBench reveals that general embedding models fall substantially short of domain-specific ones, yet training across tasks yields significant positive transfer, demonstrating that multi-task unification is not only practical but beneficial. Realizing this potential, however, requires a holistic approach: designing a unified evaluation framework, acquiring structured multi-task training data at scale, tailoring loss objectives and architectures to the resulting data structure, and thoroughly validating generalization, cross-task transfer, and robustness. We present such an end-to-end study, with the following contributions:

1. **WorkBench**, the first unified multi-task ranking benchmark for LMI, spanning six tasks curated from real-world ontologies and human-annotated resources (Section 3). WorkBench is built on publicly available data sources,

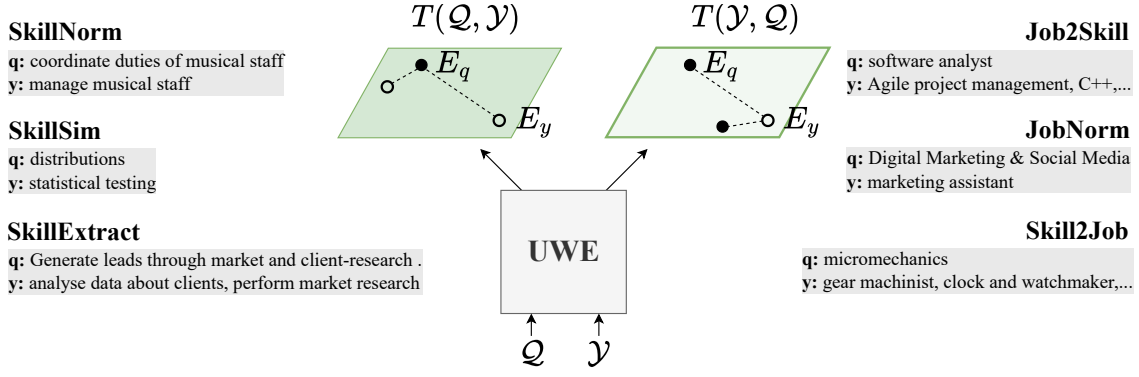


Figure 1: Overview of the six WorkBench tasks, demonstrating samples $q \in Q$ and $y \in Y$ from the query and target input spaces respectively. Unified Work Embeddings (UWE) independently encodes samples to create embeddings E_q and E_y , and maintains a bi-directional ranking structure based on embedding similarity, enabling both tasks $T(Q, Y)$ and $T(Y, Q)$. The query-target order-agnostic setup facilitates support for all WorkBench tasks, including zero-shot performance for unseen target spaces.

with the full protocol publicly released to facilitate reproducibility (Appendix A).

2. **Unified Work Embeddings (UWE)**, a 109M-parameter task-agnostic bi-encoder trained on a scalable data pipeline that fuses real-world vacancies with synthetic enrichment into structured bipartite graphs (Section 4). To exploit this structure, we introduce a *task-agnostic soft late interaction* module that unifies max-similarity (ColBERT, Khattab and Zaharia (2020)) and softmax-based (ContextMatch, Decorte et al. (2025b)) late interaction, and a *many-to-many InfoNCE* objective for symmetric multi-task learning over multi-label bipartite graphs. UWE tackles all six WorkBench tasks, generalizes zero-shot to unseen ontologies (O*NET, SkillsFuture), and outperforms the strongest specialist by +2.6 MAP and the best generalist (Qwen3-8B) by +4.4 MAP, while using $\times 73$ fewer parameters.
3. **Extensive empirical analysis** that examines cross-task transfer, loss and architecture ablations, synthetic data efficacy, cross-ontology zero-shot evaluation, and an initial gender-stereotyping bias diagnostic (Section 5).

2 Related Work

Evaluation of work-domain tasks is challenging as high-quality structured and labeled data are scarce, while large-scale unstructured data such as business data and job postings are abundant. Grounding unstructured data is key towards enabling real-world applications, by mapping noisy input data towards a unified and structured ground-truth ontology. A paramount example of

such an ontology is the multilingual classification of European Skills, Competences, and Occupations (ESCO) (le Vrang et al., 2014), driving many downstream evaluations due to its extensive and hierarchical label space of 3k job titles¹ and 14k skills. Common tasks are mapping job titles and skills to canonical examples (Bekkerman and Gavish, 2011; Javed et al., 2015; Decorte et al., 2021), predicting job titles for a given skill and vice versa (Gasco et al., 2025), extracting skills from unstructured sentences (Sayfullina et al., 2018; Zhang et al., 2022; Senger et al., 2024), and retrieving semantically similar skills (Decorte et al., 2024). Recent efforts add multilingual occupation linking benchmarks (Retyk et al., 2024) and synthetic data generation frameworks (Magron et al., 2024). As these benchmarks mainly assess task-specific models in isolation, a substantial gap remains in holistic evaluation and methodologies for the work domain. **Evaluating embedding models** with benchmarks such as BEIR (Thakur et al., 2021) and MTEB (Muennighoff et al., 2023) has driven progress in both information retrieval and generalist embedding models. While cross-encoders excel at semantic tasks (Devlin et al., 2019), their computational overhead motivates bi-encoders that encode independently (Reimers and Gurevych, 2019), or a combination of both with late interaction of the encoders (Khattab and Zaharia, 2020). In the work domain, however, evaluation of embedding model architectures and generalist models remains unexplored.

¹ESCO formally distinguishes *occupations* (groupings of similar jobs) from *job titles* (specific labels in vacancies). We define tasks agnostic to this distinction, using “job title” for both input distributions.

3 WorkBench

3.1 Tasks

WorkBench comprises six real-world work-related tasks in English for holistic evaluation in the work domain. Each task $T(Q, \mathcal{Y}) = R$ maps query space Q to target space $\mathcal{Y}$, yielding ranking matrix $R \in \mathbb{R}^{|Q| \times |\mathcal{Y}|}$. This uniform formalization enables multi-task evaluation with the common metrics defined in Section 3.2. Task examples and properties are provided in Figure 1 and Table 1 respectively, with each task detailed below. Both the benchmark and UWE are English-only. While ESCO is multilingual and our formulation is language-agnostic by design, we scope this study to English for a controlled multi-task comparison, treating multilingual extension as a natural next step (Section 8).

Job2Skill with test set $T(\mathcal{J}_{ESCO}, \mathcal{S}_{ESCO})$ exploits the widely used ESCO ontology (v1.2.0) (le Vrang et al., 2014), linking 3,039 unique job titles and 13,939 skills for job title space $\mathcal{J}_{ESCO}$ and skill space $\mathcal{S}_{ESCO}$ respectively. This is a multi-label ranking task, as each job requires multiple skills. For example, the query job title ‘software analyst’ should rank its required skills higher, such as ‘Agile project management’ and ‘C++’, compared to other skills in $\mathcal{S}_{ESCO}$. For validation, we rely on a held-out subset of the training data, based on vacancy job titles, enriched by tagged ESCO skills in the vacancy description.

Skill2Job is defined as the inverted Job2Skill task $T(\mathcal{S}_{ESCO}, \mathcal{J}_{ESCO})$ for both test and validation sets, and is a multi-label task with skills occurring in multiple job profiles. A query skill such as ‘micromechanics’ should highly rank the job titles requiring it, such as ‘gear machinist’ and ‘clock and watchmaker’.

SkillNorm with $T(\mathcal{A}_{ESCO}, \mathcal{S}_{ESCO})$ exploits ESCO’s alternative label structure where each skill in $\mathcal{S}_{ESCO}$ is provided with several alternative formulations, which we define as $\mathcal{A}_{ESCO}$. Defining the task to predict the canonical skill for each alternative enables a fixed target output space independent of the number of alternatives. For instance, the alternative formulation ‘coordinate duties of musical staff’ should retrieve its canonical ESCO skill ‘manage musical staff’.

SkillExtract relies on the SkillSpan (Zhang et al., 2022) *House* and *Tech* subsets, enriched by subsequent work with fine-grained ESCO (v1.1.0) skill labels per span (Decorte et al., 2022). We report the subtasks separately, denoted as SkillExtr-H and

SkillExtr-T respectively. The task is to predict from job vacancy sentences $\mathcal{V}$ the set of skills it contains $T(\mathcal{V}, \mathcal{S}_{ESCO})$. For the vacancy sentence ‘generate leads through market and client research’, the expressed skills ‘analyse data about clients’ and ‘perform market research’ should be ranked above all other ESCO skills.

SkillSim is a semantic skill similarity task $T(\mathcal{S}_q, \mathcal{S}_{q \cup y})$ based on the thousand related skill pairs in SkillMatch-1k (Decorte et al., 2024). A query skill such as ‘distributions’ should rank its related skill ‘statistical testing’ high, due to their semantic relatedness. As it comprises a single test dataset, we make a 10/90 split into validation and test sets. Human-annotated skill-similarity data of this quality is scarce. Therefore, the split is deliberately skewed towards the test set, reserving just enough samples for model selection while preserving a sufficiently large and reliable test set. While using the original query skills $\mathcal{S}_q$, we define an extended target space based on all 2,648 unique query and target skills $\mathcal{S}_{q \cup y}$ in the dataset, but excluding each query itself in the resulting ranking scores.

JobNorm relies on the open-sourced dataset from JobBERT (Decorte et al., 2021), containing governmental job vacancy titles $\mathcal{J}$ that are tagged with corresponding ESCO (v1.0.5) job titles $\mathcal{J}_{ESCO}$, resulting in task $T(\mathcal{J}, \mathcal{J}_{ESCO})$. We use ESCO’s canonical preferred labels as target space to maintain comparability with prior work. For example, the vacancy title ‘Digital Marketing & Social Media’ should map to the canonical ESCO job title ‘marketing assistant’.

Task	$Q \rightarrow \mathcal{Y}$	Label	Val	Test	$ \mathcal{Y} $
Job2Skill	$\mathcal{J}_{ESCO} \rightarrow \mathcal{S}_{ESCO}$	multi	5,644 [†]	3,039 [*]	13,939
Skill2Job	$\mathcal{S}_{ESCO} \rightarrow \mathcal{J}_{ESCO}$	multi	10,137 [†]	13,492 [*]	5,644 / 3,039
SkillNorm	$\mathcal{A}_{ESCO} \rightarrow \mathcal{S}_{ESCO}$	one	13,222 [*]	73,311 [*]	13,939
SkillExtr-T	$\mathcal{V} \rightarrow \mathcal{S}_{ESCO}$	multi	75 [†]	338 [†]	13,891
SkillExtr-H	$\mathcal{V} \rightarrow \mathcal{S}_{ESCO}$	multi	61 [†]	262 [†]	13,891
SkillSim	$\mathcal{S}_q \rightarrow \mathcal{S}_{q \cup y}$	one	100	900	2,648
JobNorm	$\mathcal{J} \rightarrow \mathcal{J}_{ESCO}$	one	15,462 [†]	15,461 [†]	2,942

Table 1: Overview of WorkBench query and target spaces $Q \rightarrow \mathcal{Y}$, task size for validation (val) and test set (test), ranking’s target space $\mathcal{Y}$, and multi-hot or one-hot labels. ‘[†]’ indicates only one of target or query space is ESCO-based, and ‘^{*}’ indicates that both are. For Skill2Job, $|\mathcal{Y}|$ denotes the validation / test target-space sizes, with held-out vacancy job titles as validation targets and canonical ESCO job titles as test targets.

3.2 Multi-task Evaluation

WorkBench relies on two complementary information retrieval metrics. Mean Average Precision (MAP) averages the precision at each relevant item across the full ranking, capturing both order and completeness of relevant predictions, and reduces to the Mean Reciprocal Rank for single-label tasks. R-Precision@10 (RP@10) measures the fraction of relevant items in the top 10, giving a more application-oriented view of early retrieval quality (Mitra and Craswell, 2017). Both are standard information retrieval measures applied unchanged and shared across all tasks, keeping heterogeneous tasks comparable. All task types are allocated equal importance in the final multi-task metric. Therefore, metric results are first averaged per task type (in particular subsets SkillExtr-H and SkillExtr-T), and for each metric we report the final macro average over all tasks. Following MTEB (Muennighoff et al., 2023) and BEIR (Thakur et al., 2021), WorkBench provides an evaluation-only design standardizing evaluation, with open training strategy.

4 Method

4.1 Task-agnostic encoding

WorkBench imposes three architectural constraints: flexibility across diverse tasks, zero-shot inference, as not all tasks have training data, and low-latency inference at label cardinalities up to 14k (Table 1). Cross-encoders inference time scales linearly with $|\mathcal{Y}|$ (Reimers and Gurevych, 2019), whereas bi-encoders encode inputs independently and have shown state-of-the-art performance on job normalization and skill extraction (Decorte et al., 2025b) but remain task-tailored. We therefore propose Unified Work Embeddings (UWE), a task-agnostic bi-encoder that embeds any $q \in \mathcal{Q}$ and $y \in \mathcal{Y}$, enabling a bi-directional ranking structure supporting both $T(\mathcal{Q}, \mathcal{Y})$ and $T(\mathcal{Y}, \mathcal{Q})$. At inference, only q is encoded online, with cached target embeddings. We next detail the graph-structured training data, a loss that exploits its structure, and a soft late-interaction module.

4.2 Obtaining Structured Training data

To obtain a unified embedding model, we require representative training data for each input space identified in Section 3: skills $\mathcal{S}$, job titles $\mathcal{J}$, vacancy sentences $\mathcal{V}$, and skill alternatives $\mathcal{A}$.

Grounding in ESCO skills. Following common practice (Malandri et al., 2025; Decorte et al.,

2025b), we ground all training data in the ESCO skill vocabulary. As skills are central to every WorkBench task, this creates a natural decomposition into three bipartite training graphs $(\mathcal{S}, \mathcal{J})$, $(\mathcal{S}, \mathcal{V})$, and $(\mathcal{S}, \mathcal{A})$ (Figure 2a).

Skill-Job data $(\mathcal{S}, \mathcal{J})$. Acquiring real-world skill-job edges follows the four-step pipeline in Figure 2b. (1) *Vacancy acquisition*: We scrape job vacancies from a public US jobboard, collecting 10k monthly postings over a year in 2023, yielding 104k semi-structured vacancies with a job title and description. (2) *Skill extraction*: Each vacancy description is processed with a state-of-the-art skill extraction model (Decorte et al., 2025b) to obtain ESCO skill pseudo-labels, resulting in 3.5M skill predictions (on average 33 per vacancy) with at least one confident prediction per vacancy. After case-agnostic deduplication with majority voting across duplicate job titles, 57k unique titles remain with skill profiles capped at the top-200 by averaged prediction confidence. A 90/10 split provides training and validation sets. (3) *Synthetic minority enrichment*: The resulting skill distribution (Figure 2c) is extremely long-tailed, with 862 ESCO skills absent entirely. To address this, we reverse the collection by generating job titles for underrepresented skills using GPT4o-mini. (4) *Job normalization merge*: Generated titles are matched to real vacancy titles via a job normalization model (Decorte et al., 2025b), conditionally merging their skill profiles to enrich sparse representations (details in Appendix D). This grows the dataset from 1.8M to 3.3M job-skill pairs.

Skill-Vacancy data $(\mathcal{S}, \mathcal{V})$. We rely on the synthetic dataset from prior work (Decorte et al., 2023) with 10 synthetically generated vacancy sentences per ESCO skill (v1.1.0), covering 97% of ESCO v1.2.0 skills. We add the official ESCO description for all skills to ensure full coverage.

Skill-Alternative data $(\mathcal{S}, \mathcal{A})$. We use GPT4o-mini (Achiam et al., 2023) for controlled generation of alternative skill formulations, providing three representative examples from the SkillNorm validation set to generate 20 alternatives per ESCO skill. Full details are in Appendix D, with contributions of synthetic components quantified in the ablation study (Section 6.2).

Training data quality. Because the $(\mathcal{S}, \mathcal{J})$ supervision chain combines a pseudo-labeler with LLM-generated long-tail enrichment, we audit the two stages whose outputs are not directly observed in a gold resource: (i) the merged training-graph

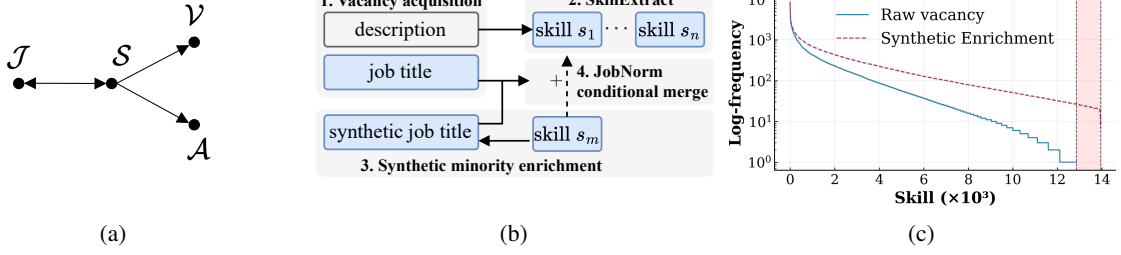


Figure 2: Overview of the training data. (a) Data dependency graph of the training data, showing the many-to-many and one-to-many edges from skill space $\mathcal{S}$ to job title $\mathcal{J}$, vacancy sentences $\mathcal{V}$, and skill alternatives $\mathcal{A}$ spaces. (b) Acquisition overview of structured training data with job titles and grounded skills. Blue colored boxes indicate structured fields. (c) Log-frequencies for the skills in the raw vacancy training dataset, before and after synthetic enrichment, with highlighted background in red indicating the gap of represented skills.

pairs (*Audit A*) and (ii) the synthetic enrichment edges that target under-represented skills (*Audit B*). A human annotator judges stratified samples from each stage against a pre-registered error taxonomy, scoring training-relevant precision that credits domain-related but non-canonical labels as useful contrastive signal. Each audit comprises 250 judgments. Audit A finds 75.2% pooled precision, rising with skill frequency from tail (60.2%) to mid-frequency (81.1%) to high-frequency (88.1%), and the training graph is itself dominated by head and mid pairs. Audit B finds 85.8% of synthetic enrichment edges plausible. The synthetic-data ablation (Section 6.2) corroborates these audits: even without synthetic enrichment, UWE (38.5 MAP) still outperforms the strongest baseline (37.6 MAP), demonstrating robustness to pipeline noise and the merits of synthetic data. Full protocol, taxonomy, and per-bucket tables are in Appendix D.5.

4.3 Many-to-many InfoNCE over bipartite graphs

Designing our multi-task training data structure, we find a highly interconnected structure as depicted in Figure 2a, where skills exhibit many-to-many edges with the job titles, and one-to-many edges with vacancy sentences and skill variants. As we will show in the following, traditional InfoNCE objectives fall short in this scenario.

Assume a mini-batch B is constructed as a sequence of query-target pairs (q, y_+) , where the query is a skill $q \in \mathcal{S}$ and the target is a matching positive from any of the remaining spaces $y_+ \in \mathcal{J} \cup \mathcal{V} \cup \mathcal{A}$. Let Y be the set of all targets in the batch, then the InfoNCE loss (Oord et al., 2018) (also known as NT-Xent (Chen et al., 2020))

for query q is

$$\mathcal{L}_{\text{InfoNCE},q} = -\log \frac{\exp(\text{sim}(q, y_+)/\tau)}{\sum_{y \in Y} \exp(\text{sim}(q, y)/\tau)}, \quad (1)$$

where $\text{sim}(q, y) = f_\theta(q) \cdot f_\theta(y)$ denotes dot product similarity of the samples encoded by model f_θ , and τ is the scaling temperature.

InfoNCE loss over a mini-batch of stacked skill-job, skill-vacancy, and skill-alias pairs has three limitations: (i) it treats each query-target pair (q, y_+) independently, regarding other-pair targets as in-batch negatives even when the same skill recurs elsewhere in the batch; (ii) it entangles all tasks into a single loss, making it sensitive to task imbalance; and (iii) it is asymmetric, ignoring many-to-many relations (a skill maps to many jobs, and a job to many skills). Given the centrality of $\mathcal{S}$ in our data structure (Figure 2a), we exploit the induced bipartite graphs $(\mathcal{S}, \mathcal{J})$, $(\mathcal{S}, \mathcal{V})$, $(\mathcal{S}, \mathcal{A})$ to address these three shortcomings with a new many-to-many (MTM) InfoNCE loss, derived next.

We first define an asymmetric one-to-many loss on a single bipartite graph. Given a mini-batch of N query-target pairs, let $G = (Q \cup Y, D)$ be a bipartite graph with $Q = \{q_1, \dots, q_N\} \subseteq \mathcal{Q}$ the query nodes, $Y = \{y_1, \dots, y_N\} \subseteq \mathcal{Y}$ the target nodes, and $D \subseteq Q \times Y$ the observed edges.

Let $Y_q = \{y \in Y \mid (q, y) \in D\}$ denote the *target set* for query q , with $|Y_q| \geq 1$ by construction. Following Khosla et al. (2020) in averaging InfoNCE outside the log for superior performance with multi-label targets, we apply node-wise averaging over queries and their target sets:

$$\mathcal{L}_{Y|Q} = -\mathbb{E}_{\substack{q \sim Q \\ y_+ \sim Y_q}} \log \frac{\exp(\text{sim}(q, y_+)/\tau)}{\sum_{y \in Y} \exp(\text{sim}(q, y)/\tau)}. \quad (2)$$

While this is a one-to-many objective over the query nodes Q , the same objective can be applied over the target nodes Y , creating the contrastive many-to-many objective:

$$\mathcal{L}_{Y,Q} = \mathcal{L}_{Y|Q} + \mathcal{L}_{Q|Y}. \quad (3)$$

Here, $\mathcal{L}_{Q|Y}$ swaps the roles of queries and targets in Eq. 2. Analogous to Y_q , each target y has a positive query set $Q_y = \{q \in Q \mid (q, y) \in D\}$, and the expectation runs over $y \sim Y$ and $q_+ \sim Q_y$, contrasting each pair against all query nodes in Q .

Having a loss for a single bipartite graph, the final loss is the weighted sum over all bipartite graphs in our data structure:

$$\mathcal{L}_{\text{MTM}} = \alpha_J \mathcal{L}_{S,J} + \alpha_V \mathcal{L}_{S,V} + \alpha_A \mathcal{L}_{S,A} \quad (4)$$

where instead of a single skill-centered graph, the separability in bipartite graphs allows controlled importance weighing with $(\alpha_J, \alpha_V, \alpha_A)$ over subsets of the multi-task data structure.

Each modification adapts InfoNCE to a property of work data the standard objective does not model: bidirectionality for many-to-many skill-job relations, averaging outside the log for multi-label targets, and per-graph separability for cross-task structure. The ablation study in Section 6.2 isolates each of these three properties, with the combined $\mathcal{L}_{\text{MTM}}$ improving task-average MAP by +1.93 over standard InfoNCE.

4.4 Task-Agnostic Soft Late Interaction

Cross-encoders score each query-target pair jointly with full token-to-token attention, yielding high accuracy but requiring one forward pass per pair. Bi-encoders encode q and y independently into a single pooled vector each, which is far more efficient but less expressive. Late-interaction architectures such as ColBERT (Khattab and Zaharia, 2020) bridge the two: each side is still encoded independently, yet scoring is performed over the full token-level matrices rather than pooled vectors, recovering much of the cross-encoder’s accuracy. While ColBERT focuses on sparse interaction by only looking at maximally similar token pairs, ContextMatch (Decorte et al., 2025b) demonstrates that a softmax-based alternative with token-level information on the sentence side markedly improves skill extraction. However, the latter is task-specific to skill extraction, while we require a holistic architecture for all WorkBench tasks. We therefore generalise both into a single, task-agnostic

module with two-sided token-level interaction and SoftMax-based fusion.

Given query q and target y with token sequences $\langle t_1, \dots, t_n \rangle$, $\langle u_1, \dots, u_m \rangle$, a shared encoder f_θ produces $E_q \in \mathbb{R}^{n \times h}$ and $E_y \in \mathbb{R}^{m \times h}$ (h : embedding dim). Define pairwise token similarities $S = E_q E_y^T \in \mathbb{R}^{n \times m}$ and $\hat{S} = \hat{E}_q \hat{E}_y^T$ for L2-normalized rows. Instead of MaxSim’s sparse binary matrix, we use temperature-controlled row-wise softmax attention:

$$A = \text{SoftMax}(S/\tau_a), \quad \tau_a > 0,$$

followed by a Frobenius inner product

$$\text{sim}(q, y) = \langle A, \hat{S} \rangle_F.$$

This *soft* aggregation provides a gradient for all token-to-token interactions with controllable smoothing. In the limit $\tau_a \rightarrow 0$, the softmax collapses to a one-hot over the most similar token pair and subsumes ColBERT’s MaxSim. With one-sided token-averaging on E_y , it subsumes ContextMatch’s mean-pooling.

5 WorkBench results

5.1 Baselines

We compare against task-specific bi-encoders **JobBERT-v2** (job normalization) and **ContextMatch** (skill extraction) (Decorte et al., 2025b), both based on the **MPNet** backbone (Song et al., 2020), which is also UWE’s base model and which we include as a domain-agnostic baseline. Both specialists train on in-domain data closely matching ours: ContextMatch uses the identical synthetic skill-sentence dataset (Decorte et al., 2023) forming our $(\mathcal{S}, \mathcal{V})$ graph, and JobBERT-v2 trains on US vacancy job titles with ESCO skills pseudo-labeled by the same extraction model as our $(\mathcal{S}, \mathcal{J})$ pipeline. For generalist models, we evaluate state-of-the-art and substantially larger models, to exemplify the gap between the best available models and a domain-tailored, more efficient one: **Qwen3** (0.6B, 4B, 8B) (Zhang et al., 2025), **E5-mistral-7B** (Wang et al., 2024), and **EmbeddingGemma** (Vera et al., 2025), all top-performing on MTEB (Muennighoff et al., 2023) with open weights. We add no prompt engineering or instruction tuning, keeping default prompts for each zero-shot generalist. Table 3 summarizes model capacity.

	Job2Skill		Skill2Jobs		JobNorm		SkillNorm		SkillSim		SkillExtr-T		SkillExtr-H		Task Avg	
	MAP	RP@10	MAP	RP@10	MAP	RP@10	MAP	RP@10	MAP	RP@10	MAP	RP@10	MAP	RP@10	MAP	RP@10
UWE (Ours)	17.9	37.8	37.3	49.4	39.3	60.8	88.9	95.9	10.4	33.8	52.3	73.5	42.3	64.2	40.2	57.7
<i>Task-specific models</i>																
ContextMatch	14.6	34.5	32.2	43.3	36.8	55.3	86.5	94.3	7.7	24.6	53.6	73.9	42.2	65.8	37.6	53.7
JobBERT-v2	0.4	0.3	0.5	0.4	39.0	58.5	83.0	91.8	9.7	30.4	35.1	48.5	20.8	38.1	26.8	37.4
<i>Gen. emb. models</i>																
MPNet	11.8	29.0	29.0	40.3	32.9	50.8	84.7	93.2	10.1	30.7	31.2	49.8	19.3	36.7	32.3	47.9
Emb. Gemma	12.6	30.1	29.9	41.6	36.7	54.6	85.1	93.9	9.3	28.7	38.1	59.3	27.3	44.9	34.4	50.2
E5-mistral-7B	11.1	28.7	29.2	40.4	35.3	54.4	83.4	92.9	12.2	37.7	32.0	45.8	19.6	33.7	32.8	49.0
Qwen3-0.6B	11.7	29.1	28.3	39.7	30.1	49.5	81.7	92.2	10.4	32.7	35.1	54.6	25.7	41.8	32.1	48.6
Qwen3-4B	13.0	31.6	31.9	43.8	33.6	53.7	85.1	94.1	12.3	37.9	37.6	58.4	28.9	46.4	34.8	52.3
Qwen3-8B	13.8	32.7	32.7	44.4	35.3	55.5	85.7	94.5	13.0	40.8	39.1	61.0	29.1	49.0	35.8	53.8

Table 2: WorkBench results (MAP and RP@10, %) for UWE, task-specific, and generalist embedding models. Task Avg is the macro-average over all tasks. Best result per column in **bold**. The state-of-the-art generalist models show that increased capacity does not close the performance gap with domain-tailored UWE.

5.2 Experimental setup

UWE uses the MPNet backbone (Song et al., 2020), with training grounded in ESCO skills through uniform sampling, yielding zero-shot predictions for Skill2Job, SkillSim, and JobNorm (Section 6.3). Training runs over 3.3M job-skill edges, with batches of 512 skills and 512 samples per $\mathcal{J}, \mathcal{V}, \mathcal{A}$ space (2,048 texts total). Vacancy sentences are augmented with random $\mathcal{V}$ samples as prefix or suffix (Decorte et al., 2025b). We use a linear schedule with warm-up rate 0.1 and peak learning rate $8e^{-5}$, selecting the best checkpoint by task-average MAP on validation sets every 50 updates, which mitigates selection instability from small validation splits such as SkillSim. For latency and FLOPS results, we benchmark each model in isolation: encoding a single query plus similarity against pre-cached targets (bf16, batch 1), reporting mean $\pm$ SE over WorkBench tasks. All experiments use an NVIDIA A100-SXM4-40GB GPU, with full setup and hyperparameters in Appendix C.

Splits and pair-level leakage. Because UWE trains and evaluates on related $(\mathcal{J}, \mathcal{S}, \mathcal{V}, \mathcal{A})$ graphs, a natural concern is whether headline gains reflect seen training distributions rather than genuine transfer. We therefore audit every WorkBench test set for leakage, defined as a held-out pair (q^*, y^*) whose canonical edge appears as a positive in any training bipartite graph. Leakage is $\leq 2.56\%$ on Job2Skill / Skill2Job and 0% on every other task, with inter-test pair overlap ≤ 8 across all task pairs (Appendix D.4). Splits are entity-disjoint by construction: the $(\mathcal{J}, \mathcal{S})$ training graph aggregates 2023 ESCO-tagged vacancies by lowercase title (skills retained at $\geq 50\%$ vacancy-level occurrence, top-200 by confidence) and is split 90/10

over unique titles. The SkillNorm task holds out one alternative per ESCO skill, and the remaining test sets are released artifacts not consumed as training pairs.

5.3 Results and Discussion

Table 2 reports UWE and all baselines across WorkBench tasks. UWE outperforms the best task-specific model ContextMatch by +2.6 MAP and +4.0 RP@10, and the best generalist Qwen3-8B by +4.4 MAP and +3.9 RP@10 on average. The in-domain grounding is measured directly by UWE against its own base model MPNet, at +7.9 MAP on an identical backbone and inference budget, while Qwen3-8B at $73\times$ the parameters recovers only 3.5 of those points. In task-specific results, UWE is state-of-the-art on Job2Skill, Skill2Jobs, and SkillNorm. For the JobNorm task, UWE outperforms the task-tailored JobBERT-v2 by +2.3 MAP and +2.3 RP@10. ContextMatch retains state-of-the-art results on SkillExtr-T, but is outperformed by UWE in MAP on SkillExtr-H. ContextMatch’s second-best overall performance suggests that skill extraction training provides positive cross-task transfer, confirmed by our analysis in Section 6.1. Generalist models improve on SkillSim with model scale, yet UWE matches Qwen3-0.6B with $5.5\times$ fewer parameters.

Table 3 reports model capacity, embedding-dimensionality, and task-averaged latency and FLOPS per query. UWE outperforms all baselines on average ranking while having nearly two orders of magnitude fewer parameters than Qwen3-8B. The MPNet-based models (UWE, ContextMatch, JobBERT-v2) show the lowest latency and FLOPS/query. Token-level late interaction raises FLOPS/query (ContextMatch $1.57 \pm 0.35\text{B}$, UWE

$3.01 \pm 0.94\text{B}$) but preserves real-time latency, remaining at least $\times 3$ faster than every generalist baseline.

Model	Param.	h	Latency (ms)	FLOPS
UWE (Ours)	109M ($\times 1$)	768	15.9 ± 0.6	$3.0 \pm 0.9\text{B}$
ContextMatch	109M	768	16.1 ± 0.7	$1.6 \pm 0.4\text{B}$
JobBERT-v2	109M	1,024	13.4 ± 0.2	$1.3 \pm 0.2\text{B}$
MPNet	109M	768	13.2 ± 0.2	$1.3 \pm 0.2\text{B}$
Emb. Gemma	308M ($\times 2.8$)	768	58.9 ± 0.1	$1.5 \pm 0.3\text{B}$
E5-mistral-7B	7B ($\times 65.2$)	4,096	61.6 ± 8.6	$112.0 \pm 21.0\text{B}$
Qwen3-0.6B	596M ($\times 5.5$)	1,024	46.2 ± 0.4	$5.9 \pm 1.2\text{B}$
Qwen3-4B	4B ($\times 36.7$)	2,560	60.0 ± 0.1	$48.4 \pm 9.9\text{B}$
Qwen3-8B	8B ($\times 73.4$)	4,096	60.2 ± 0.8	$92.4 \pm 19.0\text{B}$

Table 3: Model capacity (Param.), embedding dimensionality (h), and benchmarked latency and FLOPS per query (mean $\pm$ SE over WorkBench tasks).

6 Analysis

6.1 Cross-Task Transfer

The three bipartite loss components in $\mathcal{L}_{\text{MTM}}$ match closely with specific tasks in WorkBench: $\mathcal{L}_{S,J}$ explicitly matches skills and jobs (Skill2Job, Job2Skill), $\mathcal{L}_{S,V}$ focuses on vacancy sentences for skill extraction (SkillExtract), and $\mathcal{L}_{S,A}$ matches with the SkillNorm task. To understand how each loss component contributes to the tasks in WorkBench, we study them in isolation and measure cross-task Knowledge Gain, defined as the MAP delta from the base model with parameters θ_0 at initialization: $\text{MAP}_{\theta_t} - \text{MAP}_{\theta_0}$. For fair comparison with the base model that is not trained on token-level embeddings, all variants employ mean-pooled cosine similarity over 300 steps.

Figure 3 shows the loss components trained in isolation on the y-axis, and the WorkBench tasks on the x-axis. As expected, $\mathcal{L}_{S,J}$ improves in Job2Skill (+6.0) and Skill2Job (+7.2). However, we find it also contributes significant knowledge gain to JobNorm (+4.6), SkillNorm (+2.7) and both SkillExtract tasks (+3.9, +7.0). $\mathcal{L}_{S,V}$ shows a similar pattern, and improves the most on its matching SkillExtract tasks. $\mathcal{L}_{S,A}$ improves most for alternative skills in SkillNorm (+2.9), has positive gain on 3 tasks, and shows negative transfer for JobNorm and SkillExtr-T. Pair-level leakage is audited in Appendix D.4 and does not explain the observed transfer. Combining all three objectives reverses the per-objective negative transfer on SkillSim (Table 2), suggesting a regularization effect from multi-task training. The opposite occurs, suggesting a regularization effect: jointly

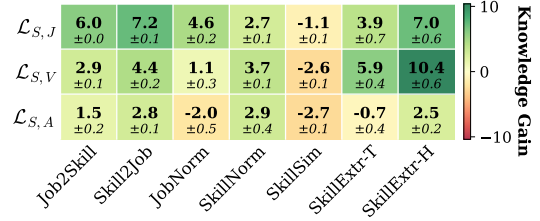


Figure 3: Cross-task transfer: partial MTM-loss objectives trained in isolation, starting from the base MPNet model. Knowledge Gain is the MAP delta, averaged over 5 runs ($\pm$ SE).

Loss	Sym.	Multi-Label	Struct.	MAP
$\mathcal{L}_{\text{InfoNCE}}$	✗	✗	✗	36.40 ± 0.07
$\mathcal{L}_{\text{MTM-pairwise}}$	✓	✗	✓	38.17 ± 0.18
$\mathcal{L}_{\text{MTM}}$	✓	✓	✓	38.33 ± 0.24

Table 4: Ablation of the MTM loss, evaluating its symmetry (sym.), multi-label support (multi-label), and structure awareness (struct.); reporting MAP (95% CI over 5 runs).

optimizing the three objectives may counteract the drift each induces towards its own input space.

6.2 Ablation Study

We ablate the loss and soft late-interaction components over 600 updates without early stopping, reporting task-average MAP on WorkBench over 5 runs (Appendix C). Table 4 ablates the three properties of $\mathcal{L}_{\text{MTM}}$ (symmetry, multi-label aggregation, and bipartite-graph structure; Eqs. 2-4) against the standard $\mathcal{L}_{\text{InfoNCE}}$ (Eq. 1, 36.40 MAP). $\mathcal{L}_{\text{MTM}}$ improves performance by +1.93 on average, whereas omitting multi-label aggregation ($\mathcal{L}_{\text{MTM-pairwise}}$) costs 0.16.

Table 5 (left) ablates the soft late interaction by comparing four aggregation choices: mean-pooled sentence embeddings ($\bar{E}_q, \bar{E}_y$), sparse token-pair MaxSim, one-sided softmax with target-side pooling ($\bar{E}_y$), and our two-sided softmax over the full token-level matrices (Reimers and Gurevych, 2019; Khattab and Zaharia, 2020; Decorte et al., 2025b). Despite operating on token-level representations, MaxSim (Khattab and Zaharia, 2020), with a sparse binary interaction matrix, is not guaranteed to outperform mean-pooled sentence embeddings on WorkBench. In contrast, both SoftMax approaches yield significant gains, with token-level representations on *both* sides achieving the best accuracy. Token-to-token interaction increases inference FLOPS by $\times 1.9$ while remaining real-time ($\approx 2.5\text{ms}$ added latency, Table 3). Appendix C.1

reports all hyper-parameter values and gridsearch ranges, including sensitivity analysis to τ_a .

For the training data pipeline, Table 6 quantifies the synthetic data components. Removing synthetic alternatives (w/o alt. synth.) reduces task-average MAP by 0.3, and further removing all synthetic enrichment (w/o all synth.) yields a 1.7 drop. Per-task results are reported in Appendix B.2.

Late Interaction	MAP
sentence emb. ($\bar{E}_q, \bar{E}_y$)	38.00 ± 0.41
MaxSim	37.63 ± 0.52
SoftMax ($\bar{E}_y$)	39.12 ± 0.10
SoftMax	39.45 ± 0.20

Table 5: Soft late interaction ablation: task-averaged MAP (95% CI, 5 runs). Methods use token-level embeddings unless marked with $\bar{E}_q$ or $\bar{E}_y$, denoting a mean-pooled query or target side. Best in **bold**.

	MAP	RP@10
UWE	40.2	57.7
w/o alt. synth.	39.9	57.1
w/o all synth.	38.5	55.9

Table 6: Synthetic data ablation: removing synthetic skill alternatives (w/o alt. synth.) and all synthetic enrichment (w/o all synth.). Best in **bold**.

6.3 Zero-Shot Generalization

To substantiate UWE’s task-agnostic performance, we evaluate zero-shot generalization at three levels: (i) *unseen target space*, where Skill2Job targets canonical ESCO occupation labels $\mathcal{J}_{\text{ESCO}}$ absent from training; (ii) *unseen task*, where JobNorm and SkillSim have no matching training objective and SkillMatch-1k (Decorte et al., 2024) queries differ in surface form from ESCO anchors; and (iii) *unseen ontology*, with query and target from O*NET (v30.1) (Tsacoumis and Willison, 2010) or the Singapore SkillsFuture (SSF) Skills Framework (Loh and Yeo, 2024). Table 7 reports cross-ontology results, where UWE outperforms the strongest baseline by +8.9 MAP on O*NET and +2.1 on SSF, confirming generalization beyond the ESCO training distribution.

6.4 Gender-Stereotype Sensitivity

As an initial diagnostic, we probe gender-stereotype associations with the Sentence Encoder Association Test (SEAT) (May et al., 2019) on SEAT-C6b for career/family concepts (Caliskan

Model	O*NET			SSF		
	Job2Skill	Skill2Job	Avg	Job2Skill	Skill2Job	Avg
UWE (Ours)	32.9	37.2	35.1	13.0	26.2	19.6
ContextMatch	20.3	32.0	26.2	11.1	22.2	16.7
Qwen3-8B	18.0	33.3	25.7	11.4	23.7	17.5

Table 7: Cross-ontology zero-shot MAP on O*NET and SkillsFuture (SSF), neither seen during training. Full results with RP@10 in Appendix B.3.

et al., 2017) and a domain-tailored test over ESCO occupations in engineering and creative arts. Both MPNet and UWE show statistically significant associations ($p < 0.01$), indicating that work-domain fine-tuning does not mitigate inherited stereotypes. Methodology and effect sizes are in Appendix B.1, with scope in Section 8.

7 Conclusion

This work introduces a first unified framework for multi-task embeddings in labor market intelligence, contributing WorkBench, an evaluation suite spanning six work-domain tasks, and Unified Work Embeddings (UWE), a bi-encoder tackling all tasks with a single model. Our analysis reveals significant cross-task transfer, with UWE outperforming the strongest specialist by +2.6 MAP and the best generalist (Qwen3-8B) by +4.4 MAP, while using two orders of magnitude fewer parameters and enabling zero-shot generalization.

8 Limitations

Language scope. WorkBench and UWE are developed and evaluated exclusively on English data. While the underlying architecture and training methodology are language-agnostic, we intentionally focus on English to establish a rigorous multi-task foundation. This scope enables controlled experimentation and principled ablations without confounding effects from cross-lingual transfer. Nevertheless, real-world labor market applications are inherently multilingual. Hence, we encourage future work to extend WorkBench along the multilingual axis, leveraging multilingual encoders (Decorte et al., 2025a) and ontologies such as ESCO’s multilingual structure (le Vrang et al., 2014).

Model usage and bias mitigation. Given the breadth of evaluations and model contributions in this work, our gender-association analysis (Section 6.4 and Appendix B.1) is intentionally preliminary, serving as an initial diagnostic rather than a comprehensive audit. The SEAT-based evaluation

measures intrinsic associative structure in the embedding space, but does not directly quantify downstream impact on hiring or matching outcomes, which would require demographically annotated queries that WorkBench does not contain. Its practical role is that of a cheap pre-deployment screen computed without task labels, where a significant effect signals that a task-level audit is warranted. Therefore, we recommend task-specific audits before deployment in sensitive settings. We encourage future work to focus on extensive bias analysis across protected attributes in our proposed multi-task and zero-shot setup, and to investigate debiasing techniques tailored to multi-task work embeddings.

Acknowledgments

This research was funded by TechWolf, and VLAIO (Flanders innovation & entrepreneurship) through the Baekeland grant (HBC.2025.0886).

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Ron Bekkerman and Matan Gavish. 2011. [High-precision phrase-based document classification on a modern scale](#). In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, page 231–239, New York, NY, USA. Association for Computing Machinery.
- Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29.
- Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmlR.
- Matthias De Lange, Warre Veys, Federico Retyk, Daniel Deniz, Warren Jouanneau, Mike Zhang, Aleksander Bielinski, Emma Jouffroy, Nicole Clobes, Nina Baranowska, David Graus, Marc Palyart, Rabih Zbib, Dimitra Gkatzia, Thomas Demeester, Tijl De Bie, Toine Bogers, Jens-Joris Decorte, and Jeroen Van Haute. 2026. [Workrb: A community-driven evaluation framework for ai in the work domain](#). *Preprint*, arXiv:2604.13055.
- Jens-Joris Decorte, Matthias De Lange, and Jeroen Van Haute. 2025a. [Multilingual jobbert for cross-lingual job title matching](#). *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025)*, 4038.
- Jens-Joris Decorte, Jeroen Van Haute, Johannes Deleu, Chris Develder, and Thomas Demeester. 2022. Design of negative sampling strategies for distantly supervised skill extraction.
- Jens-Joris Decorte, Jeroen Van Haute, Thomas Demeester, and Chris Develder. 2021. [JobBERT : understanding job titles through skills](#). In *FEAST, ECML-PKDD 2021 Workshop, Proceedings*, page 9.
- Jens-Joris Decorte, Jeroen Van Haute, Thomas Demeester, and Chris Develder. 2024. Skillmatch: Evaluating self-supervised learning of skill relatedness. *arXiv preprint arXiv:2410.05006*.
- Jens-Joris Decorte, Jeroen Van Haute, Chris Develder, and Thomas Demeester. 2025b. Efficient text encoders for labor market analysis. *arXiv preprint arXiv:2505.24640*.
- Jens-Joris Decorte, Severine Verlinden, Jeroen Van Haute, Johannes Deleu, Chris Develder, and Thomas Demeester. 2023. [Extreme multi-label skill extraction training using large language models](#). In *AI4HR & PES 2023 : International Workshop on AI for Human Resources and Public Employment Services, Proceedings*, pages 1–10.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.
- Luis Gasco, Hermenegildo Fabregat, Laura García-Sardiña, Paula Estrella, Daniel Deniz, Alvaro Rodrigo, and Rabih Zbib. 2025. [Overview of the talentclef 2025: Skill and job title intelligence for human capital management](#). *Preprint*, arXiv:2507.13275.
- Faizan Javed, Qinlong Luo, Matt McNair, Ferosh Jacob, Meng Zhao, and Tae Seung Kang. 2015. [Carotene: A job title classification system for the online recruitment domain](#). In *2015 IEEE First International Conference on Big Data Computing Service and Applications*, pages 286–293.
- Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschiot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.
- Martin le Vrang, Agis Papantoniou, Erika Pauwels, Pieter Fannes, Dominique Vandenstein, and Johan De Smedt. 2014. Esco: Boosting job matching in europe with semantic interoperability. *Computer*, 47(10):57–64.
- Chelvin Loh and Wee Siong Yeo. 2024. Forward singapore: Strengthening skillsfuture. *Singapore Labour Journal*, 3(01):114–122.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Antoine Magron, Anna Dai, Mike Zhang, Syrielle Montariol, and Antoine Bosselut. 2024. [JobSkape: A framework for generating synthetic job postings to enhance skill matching](#). In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 43–58, St. Julian’s, Malta. Association for Computational Linguistics.
- Lorenzo Malandri, Fabio Mercorio, and Antonio Serino. 2025. Skillmo: Normalized esco skill extraction through transformer models. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, pages 1969–1978.
- Chandler May, Alex Wang, Shikha Bordia, Samuel Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628.
- Bhaskar Mitra and Nick Craswell. 2017. Neural models for information retrieval. *arXiv preprint arXiv:1705.01509*.
- Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. [MTEB: Massive text embedding benchmark](#). In *Proceedings of the 17th Meeting of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Federico Retyk, Luis Gasco, Casimiro Pio Carrino, Daniel Deniz, and Rabih Zbib. 2024. [MELO: An Evaluation Benchmark for Multilingual Entity Linking of Occupations](#). *The 4th Workshop on Recommender Systems for Human Resources (RecSys in HR’24)*, in conjunction with the 18th ACM Conference on Recommender Systems.
- Luiza Sayfullina, Eric Malmi, and Juho Kannala. 2018. Learning representations for soft skill matching. In *Analysis of Images, Social Networks and Texts*, pages 141–152, Cham. Springer International Publishing.
- Elena Senger, Mike Zhang, Rob van der Goot, and Barbara Plank. 2024. [Deep learning-based computational job market analysis: A survey on skill extraction and classification from job postings](#). In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 1–15, St. Julian’s, Malta. Association for Computational Linguistics.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MpNet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.
- Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. [Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.
- Suzanne Tsacoumis and Shannon Willison. 2010. O* net analyst occupational skill ratings: Procedures. Alexandria, VA: Human Resources Research Organization.
- Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, and 1 others. 2025. Embeddinggemma: Powerful and lightweight text representations. *arXiv preprint arXiv:2509.20354*.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916.
- Mike Zhang, Kristian Jensen, Sif Sonniks, and Barbara Plank. 2022. [SkillSpan: Hard and soft skill extraction from English job postings](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4962–4984, Seattle, United States. Association for Computational Linguistics.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren

Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Data and Code Availability

We release the model, the WorkBench evaluation suite, and the evaluation code to facilitate reproducibility and future research. WorkBench is constructed entirely from publicly available resources, namely the ESCO, O*NET, and SkillsFuture ontologies together with the human-annotated and human-curated test sets listed below, and is not dependent on any proprietary or institution-internal data. It is released in the continuous evolving open-source project WorkRB² (De Lange et al., 2026).

A.1 Data license overview

In accordance with the ACL Responsible NLP Research Checklist, we summarize the licenses for third-party data sources in this work:

- **ESCO ontology** (le Vrang et al., 2014) is released under the Creative Commons Attribution 4.0 license (CC BY 4.0).
- **SkillExtract benchmarks** HOUSE and TECH subsets (Decorte et al., 2022) are derived from SkillSpan (Zhang et al., 2022) with ESCO skill annotations and are released under CC BY 4.0.
- **SkillMatch-1K** (Decorte et al., 2024) used for SkillSim is released under the MIT license.
- **JobBERT evaluation dataset** (Decorte et al., 2021) used for JobNorm is released under the MIT license.

A.2 AI Writing and Coding Assistance

During the preparation of this manuscript, the authors used generative AI tools, including ChatGPT, Gemini, and Claude, for language polishing and proofreading of author-written content. Additionally, Cursor (an AI-assisted coding environment) along with the aforementioned language models were used for coding assistance, specifically for debugging, code review, and implementation of routine components. In all cases, these tools served in an assistive capacity for short-term, well-defined tasks under direct human supervision. The authors retain full responsibility for the correctness and

originality of the research ideas, methodology, experimental results, and all content presented in this work. No AI-generated text was used to produce novel research ideas or substantive written content.

B Extended Experimental Results

B.1 Preliminary Analysis of Gender-Occupation Associations

Motivation and scope. Language models trained on large-scale corpora risk encoding societal biases that propagate to downstream applications (Bolukbasi et al., 2016). For work-domain models, such biases warrant particular scrutiny as systematic associations between demographic attributes and occupations or skills could adversely affect hiring recommendations and talent matching systems.

Given the breadth of evaluations and models studied in this work, a comprehensive bias analysis for multi-task work embeddings is beyond our scope. Nevertheless, given the real-world implications of work-domain NLP, we report a preliminary sensitivity probe to characterize baseline bias properties. We adopt the Sentence Encoder Association Test (SEAT) (May et al., 2019), a standard intrinsic evaluation protocol for probing stereotypical associations in embedding spaces. Since UWE inherits its initialization from MPNet and is adapted to synthetic data that may encode implicit biases, we expect the model to exhibit detectable stereotypical associations. Based on this preliminary investigation, we encourage future work to focus on comprehensive bias auditing and mitigation for multi-task work embeddings.

Method. Following May et al. (2019), SEAT defines two *target* embedding sets X and Y of equal size (e.g., male vs. female) and two *attribute* sets A and B (e.g., career vs. family). The test statistic is the difference of sums of the target concepts

$$s(X, Y, A, B) = \left[\sum_{x \in X} s(x, A, B) - \sum_{y \in Y} s(y, A, B) \right]$$

where for the embeddings of an input sentence q , we compute an association score with the attributes

$$s(q, A, B) = \mathbb{E}_{a \sim A} [\text{sim}(q, a)] - \mathbb{E}_{b \sim B} [\text{sim}(q, b)],$$

with $\text{sim}(\cdot, \cdot)$ the model’s similarity function. UWE computes similarity via its task-agnostic soft late interaction over token-level embeddings, while

²<https://github.com/techwolf-ai/workrb>

Target concepts	Attribute concepts
<i>Male</i> 'The sons are there.', 'A male is a person.', 'Boys are people.',...	<i>Engineering</i> 'engine designer', 'food production engineer', 'environmental expert',...
<i>Female</i> 'The girls are there.', 'Here she is.', 'These are sisters.',...	<i>Arts</i> 'music director', 'sculptor', 'video artist',...

Table 8: Example target and attribute concepts for ESCO Eng/Arts (M/F), showing three instances per category.

MPNet uses cosine similarity on mean-pooled sentence embeddings. We perform the SEAT permutation test, reporting effect size (Cohen’s d) and significance at $p < 0.01$. Significant results indicate detectable bias; non-significant results should be interpreted as failure to detect bias rather than evidence of its absence.

Association tests. We evaluate two gender-association benchmarks. The first, **SEAT-C6b (Male/Female vs. Career/Family)**, is a standard benchmark from Caliskan et al. (2017) adapted to sentence encoders by May et al. (2019), measuring associations between gendered target sentences and career/family attributes.

To approximate downstream task conditions, we construct a second benchmark, **ESCO Engineering/Arts (M/F)**, using actual job titles from the work domain. Resembling implicit human association in Caliskan Test 7 and 8 (Caliskan et al., 2017), we measure the association between male and female targets, and attributes engineering and arts. Male and female target sentences are obtained from the SEAT-C6b test. Additionally, we collect job titles from ESCO (le Vrang et al., 2014), selecting ISCO prefix 214 (Engineering professionals, excluding electrotechnology; 124 titles) and prefix 265 (Creative and performing artists; 56 titles), rebalancing to equal size via random subsampling. Table 8 shows representative examples.

Results. Table 9 reports effect sizes for MPNet (UWE’s base encoder) and UWE. Both models exhibit statistically significant associations on both benchmarks, confirming gender-occupation stereotypes in the embedding space. UWE retains similar effect size to MPNet on SEAT-C6b and increased effect size on ESCO Eng/Arts.

As a secondary analysis, we investigate whether UWE’s architectural design affects measured asso-

	ESCO Eng/Arts (M/F)	SEAT-C6b
MPNet (base model)	1.160*	0.388*
UWE	1.570*	0.392*

Table 9: SEAT effect sizes for UWE and its base model. *indicates $p < 0.01$.

Late Interaction	ESCO Eng/Arts (M/F)	SEAT-C6b
Sentence embeddings	1.508*	0.342
MaxSim	1.556*	0.449*
SoftMax- y -mean	1.503*	0.389*
SoftMax	1.479*	0.503*

Table 10: SEAT effect sizes across late-interaction variants. *indicates $p < 0.01$.

ciations, specifically the introduced token-level late interaction. Table 10 evaluates the late-interaction variants from the main paper’s ablation study in Section 6.2. All variants show significant bias on ESCO Eng/Arts, while results on SEAT-C6b are more variable with sentence embeddings showing no significant effect. These preliminary findings suggest that interaction mechanism influences measured associations, though the relationship warrants further investigation.

Limitations and future work. Given the breadth of evaluations and model contributions in this work, our bias analysis is necessarily preliminary, and we hope to inspire future work to focus on extensive bias analysis in our proposed multi-task and zero-shot setup. This preliminary analysis is limited to gender; future work should examine additional sensitivity to protected attributes and social biases. Beyond analysis, we encourage future work to investigate whether established debiasing techniques transfer effectively to multi-task work embeddings.

B.2 Synthetic Data Ablation: Full Results

Table 11 reports the per-task breakdown of the synthetic data ablation summarized in Section 6.2. Removing all synthetic enrichment (w/o all synth.), includes removal of synthetic skill alternatives, synthetic job titles from inversed minority enrichment, and synthetic vacancy training data. Alternative removal alone (w/o alt. synth.) has a mixed effect across tasks, with an average decrease of -0.3 MAP and 0.6 RP@10. When all synthetic data is removed, results show this impacts performance of the related tasks directly for SkillNorm, JobNorm, and the skill extraction tasks. Additionally, the synthetic data helps to improve on other tasks

such as Job2Skill, SkillsSim, and with the largest improvement on Skill2Job.

B.3 Cross-Ontology Zero-Shot: Full Results

Table 12 reports the full cross-ontology zero-shot results with both MAP and $RP@10$ on O*NET and SkillsFuture (SSF), complementing the MAP-only results in Table 7. UWE significantly outperforms all baselines on both ontologies across both metrics: +8.9 MAP and +11.1 $RP@10$ on O*NET, and +2.1 MAP and +1.9 $RP@10$ on SSF, over the respective best baseline.

C Experiment Setup Details

C.1 Hyper-parameter Search

To obtain an experimental configuration, we first perform a gridsearch of the hyper-parameters, evaluated on task-average MAP performance after 600 update steps. We denote the best setting in bold in the following.

- **Soft Late Interaction temperature** τ_a is the temperature used for scaling of the SoftMax to obtain the interaction matrix A . We perform a gridsearch over $\{2.0, 1.0, \mathbf{0.5}, 0.1\}$.
- **Per-task loss components** are α_J, α_V , and α_A . For a gridsearch over values $\{0, 0.5, 1.0\}$, we find configuration $\alpha_J = 1.0, \alpha_V = 0.5, \alpha_A = 0.5$.
- **Learning rate** for the AdamW optimizer is $\{1 \times 10^{-4}, \mathbf{8 \times 10^{-4}}, 5 \times 10^{-5}, 1 \times 10^{-5}\}$.
- **MTM loss temperature** τ in Eq. 2 has a gridsearch over $\{0.01, 0.02, 0.05, 0.1\}$ for the 3 loss components, resulting in 0.05 for $\mathcal{L}_{S,J}$, and 0.02 for $\mathcal{L}_{S,V}, \mathcal{L}_{S,A}$.

The following configurations are kept constant:

- **Base encoder:** all-mpnet-base-v2 (109M parameters)³, a state-of-the-art MPNet-based model (Song et al., 2020).
- **Batch composition:** 512 skills, 512 job titles, 512 vacancy sentences, 512 skill alternatives (2,048 texts per update step).
- **Optimiser and learning rate schedule:** We use the AdamW optimizer (Loshchilov and

Hutter, 2017) with a learn learning rate schedule where warm-up uses the first 10% of training length, followed by linear decay. Peak learning rate is defined by the gridsearch above.

- **Token budget:** To constrain memory size during training, we set a max token capacity of 64 for all inputs, with no constraints during inference. We find this budget sufficient to support the sentence length of all WorkBench tasks
- **Hardware:** All experiments are conducted on a NVIDIA A100-SXM4-40GB GPU.

C.2 Experiment Details

C.2.1 Loss ablation.

Results in Table 4 of the main paper use the best setup defined in the hyper-parameter search. However, to disentangle the effects of using our soft late interaction, we apply the standard mean pooling of the embeddings instead. The final results are averaged over 5 runs over different seeds of 600 update steps each, over which we report the 95% confidence interval.

C.2.2 Late interaction ablation.

Results in Table 5 of the main paper also use the best setup defined in the hyper-parameter search. Similar to the previous setup, the final results are averaged over 5 runs over different seeds of 600 update steps each, over which we report the 95% confidence interval. Figure 4 shows an overview of influence of the temperature τ_a on the SoftMax objectives, where MaxSim is depicted as having $\tau_a = 0$. The results show that the token-level Softmax outperforms the target-averaged version, but only for a lower temperature. After the gridsearch, the best setting is run for the 5 seeds, resulting in $\tau_a = 0.1$ for token-level SoftMax (*Softmax - token*), and $\tau_a = 0.5$ for target-averaged embeddings (*Softmax - mean*).

C.2.3 Cross-task transfer.

To obtain results in Figure 3 in the main paper, we make a reduced 300 optimisation steps per objective due to faster convergence of the single-component losses. The experiment uses the best setup defined in the hyper-parameter search. For each of $\mathcal{L}_{S,J}, \mathcal{L}_{S,V}, \mathcal{L}_{S,A}$, we make 5 runs over different seeds and report standard error (SE). We

³<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

	Job2Skill		Skill2Job		JobNorm		SkillNorm		SkillSim		SkillExtr-T		SkillExtr-H		Task Avg	
	MAP	RP@10	MAP	RP@10	MAP	RP@10	MAP	RP@10	MAP	RP@10	MAP	RP@10	MAP	RP@10	MAP	RP@10
UWE	17.9	37.8	37.3	49.4	39.3	60.8	88.9	95.9	10.4	33.8	52.3	73.5	42.3	64.2	40.2	57.7
w/o alt. synth.	17.7	37.5	37.0	49.2	39.6	60.7	87.9	95.3	9.9	32.3	51.8	73.3	42.6	62.4	39.9	57.1
w/o all synth.	17.7	37.6	35.9	47.7	37.7	57.4	87.7	94.9	10.3	32.4	46.0	69.6	37.5	61.2	38.5	55.9

Table 11: Full per-task results of the synthetic data ablation on WorkBench, reporting MAP and RP@10. “w/o alt. synth.” removes synthetic skill alternatives; “w/o all synth.” removes all synthetic enrichment in the training data.

	Job2Skill		Skill2Job		Task Avg	
	MAP	RP@10	MAP	RP@10	MAP	RP@10
<i>O*NET</i>						
UWE (Ours)	32.9	39.3	37.2	57.0	35.1	48.1
ContextMatch	20.3	21.9	32.0	51.1	26.2	36.5
Qwen3-8B	18.0	20.7	33.3	53.4	25.7	37.0
<i>SkillsFuture (SSF)</i>						
UWE (Ours)	13.0	20.2	26.2	32.3	19.6	26.3
ContextMatch	11.1	18.4	22.2	28.4	16.7	23.4
Qwen3-8B	11.4	19.3	23.7	29.5	17.5	24.4

Table 12: Full cross-ontology zero-shot results (MAP and RP@10, %) on O*NET and SkillsFuture (SSF). Neither ontology is seen during training. Best result per column in **bold**.

calculate the Knowledge Gain based on the average per-task performance of the final model and the base MPNet model from which training was initiated.

C.2.4 Timing evaluation.

To obtain the latency and FLOPS per query in Table 3 in the main paper, we run each baseline in isolation. To measure the metrics at inference, we cache the model target space embeddings, and make a measurement for each query in a subset per task. We enable a warmup of 30 queries to avoid additional latencies on initial memory allocations, then we measure 100 queries per task to measure la-

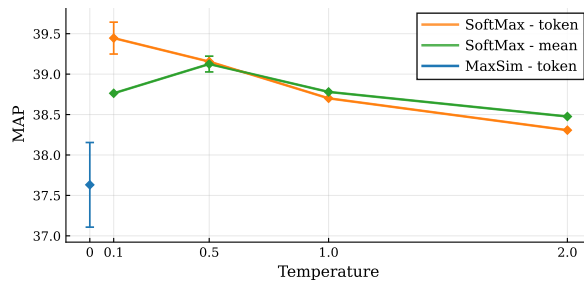


Figure 4: Overview of temperature τ_a influence for late interaction, including SoftMax with token-level embeddings (token), and target-averaged embeddings (mean). MaxSim is depicted as having temperature zero. The final results after the gridsearch are depicted with 95% CI.

tencies, and 10 separate queries per task to measure the FLOPS. We allocate more budget to latencies due to their high variability. The final reported results first average per task over all queries, and then calculate the macro-average on WorkBench with standard error over tasks.

C.2.5 Final WorkBench evaluation.

The results in Table 2 in the main paper rely on the publicly available models for each of the baselines. The task-specific models JobBERT-v2 and ContextMatch are additionally verified to have matching results from their original JobNorm, and SkillExtr-H and SkillExtr-T benchmarks (Decorte et al., 2025b). For our final model, we rely on the hyperparameters as described in the grid-search above. Furthermore, following Decorte et al. (2025b), we randomly apply a prefix or suffix vacancy sentence to each of the vacancy sentences. We ensure it is a non-matching sentence with any vacancy sentence in the mini-batch, by sampling a vacancy sentence for the out-of-batch skills. We ablate the sampling probability and apply a grid-search over values of $\{0.0, 0.8\}$. In contrast to the ablation experiments, we evaluate the model every 50 update steps and retain the final model with the best task-average evaluation performance.

D Data pipeline details

D.1 Inversed synthetic minority enrichment.

D.1.1 Synthetic job title generation

To obtain the generated job titles for minority skills, we generate job titles with a generative language model. Using GPT4o-mini (Achiam et al., 2023), we set a minimum number of $N = 20$ generations for each ESCO skill. This ensures all minority skills are represented by a minimal subset of job titles. Additionally, this ensures majority skills are enriched from a similar data-generating distribution as the minority skills. Using structured output, we enforce a list-like return type of the generative model. However, we found this resulted in a wide range in the number of generations per skill. Espe-

cially as we parse the results to be valid only if the generated title contains at least two words. Therefore, we enabled a retry mechanism that would retry the prompt asking for the remaining number of required job titles, and providing the already generated job titles in context. Instead of generating a single job title N times independently, this procedure ensures we have a minimal of duplicate job titles per skill by providing context to the generative model.

The prompt is formulated as follows for an initial $N = 20$ job titles to be generated, per skill s_i , with ESCO description d_i : Return a list of N unique, real-world job titles for which the given skill is required. 'Real-world' means titles actually used in industry or job ads. Make heavy use of alternative spellings, paraphrases, synonyms. Add heavy noise such as alternative upper casing, spellings, paraphrases, synonyms. Add the noisy information using comma-separation, brackets, &, @, -, or other separators. Remove exact duplicates so every title is unique. The skill can only occur explicitly in the first job title. You are given: `<skill_name> $s_i$ </skill_name>` `<skill_description> $d_i$ </skill_description>`

When a retry is applied since there are insufficiently generated job titles for the skill, N is set to the remaining number of required job titles, and the prompt is appended with the already generated job titles J (variables indicated in bold): You already provided these job titles, make sure there are no duplicates: J

D.1.2 Merging synthetic job titles into real-world data.

The resulting 285k generated job titles have sparse skill profiles with just a single skill per title, rather than the many-to-many structure present in real vacancy data. A first approach is to deduplicate the generated job titles, but this enables merging skills from only a small fraction of exact matches. Therefore, we match synthetic titles to existing real-world vacancy titles: after a match, the synthetic title's skill is added to the matched vacancy's skill profile, enriching it with minority skills. The remaining synthetic titles without matches are also retained as separate entries in the dataset, growing the total from 1.8M to 3.3M job-skill pairs.

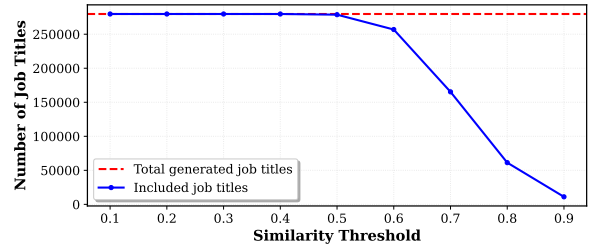


Figure 5: Job title similarity threshold over the number of included jobs. Synthetic job title skills are merged with real-world job titles when above the similarity threshold.

Using state-of-the-art model JobBERT-v2 (Decorte et al., 2025b) we find the job similarity between the synthetic and real-world job titles. A match is determined based on a similarity threshold of the cosine similarity between the embeddings of both job titles. Figure 5 depicts the total number of generated job titles, and illustrates the included job titles for a range of thresholds. After qualitative evaluation on a subset, we pick the threshold 0.7 that results in 165k out of the 285k generated titles having a match in the real-world job titles. A synthetic job title can have multiple matches with real-world job titles, as long as the similarity is above the threshold. While this work focuses on English, the procedure is extensible beyond an English setting using a multilingual encoder (Decorte et al., 2025a).

D.2 Fully synthetic data for SkillNorm.

For skill normalization, WorkBench employs the full ESCO dataset for validation and test sets. For training data, however, this task is specifically suited for synthetic data generation. To this end, we employ GPT4o-mini (Achiam et al., 2023) to generate a set of alternative formulations A for each of the ESCO skills. The prompt for skill s_i is formulated as follows, including a fixed set of 3 representative examples from the validation set: Return a LIST OF 20 UNIQUE PARAPHRASES of the given skill. Each paraphrase should express the same core meaning but use different wording. Use synonyms, change word order, add or modify verbs, make it more specific. Examples: supervise correctional procedures -> oversee prison procedures, show initiative -> initiate action lead police investigations -> manage police enquiries. Remove exact duplicates so every paraphrase is unique. Maintain the same skill

meaning while varying the expression. You are given: `<skill_name> $s_i$ </skill_name>`

D.3 WorkBench dataset characteristics

Table 1 summarizes WorkBench task sizes; here we provide a finer-grained characterization of each underlying source.

ESCO v1.2.0 (Job2Skill, Skill2Job, SkillNorm, SkillExtract target space). ESCO contains 13,939 skills partitioned by `skillType` into 10,715 *skill/competence* entries and 3,219 *knowledge* entries, and by `reuseLevel` into 6,655 sector-specific, 3,783 cross-sector, 3,044 occupation-specific, and 452 transversal skills. 97.5% of skills carry ≥ 1 alternative label, totaling 86,533 alternatives (avg. 6.4 per skill with ≥ 1), which forms the supervision for SkillNorm. The occupation pillar holds 3,039 occupation entries linked to skills through 129,004 relations (67,622 essential, 61,382 optional).

JobBERT / JobNorm. The JobBERT test set contains 15,463 unique non-canonical job titles mapped to 1,369 ESCO occupation labels (out of the 2,942-occupation target space in Table 1). The mapping is highly long-tailed: an ESCO occupation receives between 1 and 647 source titles (mean 11.3, median 3). Source titles are very short (mean 3.5 whitespace tokens), reflecting raw vacancy headers rather than full descriptions.

SkillSpan House / Tech (SkillExtract). The two subsets are drawn from different vacancy registers and differ substantially in style. *Tech* contains 338 test (75 val) sentences, mean length 12.9 tokens, with 1.99 skills/sentence on average over a 288-skill restricted vocabulary (286 used). *House* (non-tech) contains 262 test (61 val) sentences, mean length 22.6 tokens, with 2.17 skills/sentence over a 391-skill restricted vocabulary (389 used). Both subsets are projected onto the full 13,891-skill ESCO target space at evaluation time (Table 1).

SkillMatch-1k (SkillSim). The benchmark contains 2,000 human-annotated skill pairs (1,800 test + 200 val), balanced 1,000 related / 1,000 unrelated. 1,401 unique query skills appear as `skill_a`, 1,699 unique candidate skills as `skill_b`, with 2,648 distinct skills in the union. This union forms the target space $|\mathcal{Y}| = 2,648$ used in Table 1.

D.4 Train/test pair-level leakage

We audit, for every WorkBench test set, the pair-

level leakage and the inter-test pair overlap. A test pair (q^*, y^*) is non-trivial only insofar as the model has not learned this specific link during training; surface-form overlap of q^* or y^* alone with training entities is not, by itself, leakage. We canonicalize all strings with case-folding and whitespace stripping, and report results on the full training pool (real ESCO-tagged vacancies + synthetic minority enrichment, Sec 4).

Edge-level leakage: a test pair (q^*, y^*) leaks iff the canonical pair appears as a positive edge in some training bipartite graph. UWE trains on $(\mathcal{S}, \mathcal{J})$, $(\mathcal{S}, \mathcal{V})$, and $(\mathcal{S}, \mathcal{A})$, so tasks with no matching edge type (SkillSim, JobNorm, cross-ontology) are zero by construction.

Inter-test pair overlap: the same canonical pair (q^*, y^*) appearing in two distinct test sets, which would inflate any cross-task transfer claim built on shared instances.

Table 13 reports the per-task pair-level leakage. It is at most 2.56% across all tasks (Job2Skill / Skill2Job; 0.86% when restricted to the real-only training pool). For SkillNorm and SkillExtract the measured leakage is 0%: although these tasks share a training graph type with their supervision $((\mathcal{S}, \mathcal{A})$ and $(\mathcal{S}, \mathcal{V})$ respectively), the held-out test pairs are disjoint from the training pairs by construction. SkillSim and JobNorm are 0% by construction since neither $(\mathcal{S}, \mathcal{S})$ nor $(\mathcal{J}, \mathcal{J})$ is present as a training bipartite graph. Inter-test pair overlap is 0 for every WorkBench task pair except Skill2Job $\cap$ JobNorm, which shares 8 pairs (0.054% of JobNorm test pairs, 0.006% of Skill2Job test pairs). Cross-task transfer (Section 6.1) therefore cannot be attributed to memorized or shared pairs.

A weaker definition of leakage. Both endpoints of a test pair appearing as training entities (in training pairs *other* than (q^*, y^*)) could be seen as leakage for a bi-encoder, since the encoder has then seen both ends of the held-out link. Under this looser definition, Job2Skill / Skill2Job reach 14.55% (11.68% real-only), driven by the synthetic minority-skill enrichment that, by design, generates realistic occupation-like titles for ESCO skills, many of which incidentally match ESCO canonical occupation labels. For all other WorkBench tasks the measure stays $\leq 2.77\%$ (SkillExtract: $\leq 0.17\%$; SkillNorm: 1.47%; SkillSim: 1.56%; JobNorm: 2.77%). We do not consider this leakage in the strict sense (no training pair is the test

Task	Train graph	Test pairs	Leakage
Job2Skill	$(\mathcal{S}, \mathcal{J})$	128,987	2.56%
Skill2Job	$(\mathcal{S}, \mathcal{J})$ rev.	128,987	2.56%
SkillNorm	$(\mathcal{S}, \mathcal{A})$	73,300	0.00%
SkillExtr-H	$(\mathcal{S}, \mathcal{V})$	529	0.00%
SkillExtr-T	$(\mathcal{S}, \mathcal{V})$	583	0.00%
SkillSim	none	900	0.00% [†]
JobNorm	none	14,765	0.00% [†]

Table 13: Pair-level train/test leakage on WorkBench: fraction of test pairs (q^*, y^*) whose canonical edge appears as a positive in some training bipartite graph. Computed on the full training pool (real + synthetic). “†” indicates zero by construction: the task’s (q, y) edge type is not present in any UWE training graph. Inter-test pair overlap is ≤ 8 pairs across all task combinations.

pair), but report it for completeness.

D.5 Human audit of training-data quality

We pre-register two audits of the model-derived signals in Sec 4: pseudo-label precision in the merged training graph (Audit A) and synthetic injected-edge plausibility (Audit B). Pre-registration uses frozen-sample parquets with recorded random seed, source-file SHAs, and git commit; samples are committed before annotation begins. Each audit consists of 250 human judgments.

Audit A: pseudo-labels in the merged training graph. We sample (title, skill) pairs from the merged $(\mathcal{S}, \mathcal{J})$ training graph, stratified by skill rank-fraction into *head* (top 10%), *torso* (10–50%), and *tail* (50–100%). Each pair is judged CORRECT / INCORRECT / UNCERTAIN against a five-way error taxonomy: *wrong_domain* (the predicted skill is from an unrelated domain), *loosely_related* (same domain as the title, but not the canonical match), *title_too_ambiguous* (the title alone is too generic to license any specific skill, e.g. “Student Employee” $\rightarrow$ *wash the laundry*), *over_narrow_skill* (the skill is a plausible specialization but more specific than the title warrants), *over_broad_skill* (the skill is in the right area but too generic to be informative for the title).

We report two precisions. *Strict* precision counts only CORRECT; *training-relevant* precision additionally counts *loosely_related*, *over_narrow_skill*, and *over_broad_skill* as useful, since for a contrastive bi-encoder these still pull the embedding toward a related skill cluster rather than an arbitrary wrong skill. Only *wrong_domain* is a truly uninformative training signal. Table 14

reports both per bucket. The pooled rate is biased downward because the sample oversamples the tail relative to the actual training-graph distribution, which is dominated by head and torso pairs.

Bucket	n	Strict	Training-relevant
head	67	80.6%	88.1%
torso	90	58.9%	81.1%
tail	93	45.2%	60.2%
pooled	250	59.6%	75.2%

Table 14: Audit A: pseudo-label precision in the merged $(\mathcal{S}, \mathcal{J})$ training graph. Strict counts CORRECT; training-relevant additionally counts domain-related-but-non-canonical labels as useful contrastive signal, excluding only *wrong_domain* errors.

Among the 101 INCORRECT judgments, the error-type distribution is: *wrong_domain* 55.4%, *loosely_related* 33.7%, *title_too_ambiguous* 5.9%, *over_narrow_skill* 3.0%, *over_broad_skill* 2.0%. Loosely-related errors are the second-largest class, supporting the training-relevant framing.

Audit B: synthetic injected-edge plausibility.

We sample (real title, ESCO skill) edges produced by the long-tail synthetic enrichment merge step (Sec 4), stratified into *head_sanity* (top 10% skills, sanity check) and *long_tail* (the under-represented skills the enrichment is designed to address). Edges are judged PLAUSIBLE / IMPLAUSIBLE / UNCERTAIN against a five-way error taxonomy: *acceptable_but_weak*, *wrong_normalization_target* (the synthetic title was matched to the wrong real vacancy by the job-normalization step), *over_narrow_skill*, *over_broad_skill*, and *hallucinated_skill_in_synth* (the LLM generated a skill not in the prompt). Training-relevant additionally counts *acceptable_but_weak*, *over_narrow_skill*, and *over_broad_skill* as useful. Of the 250 judged edges, the 4 marked UNCERTAIN are excluded from the denominators.

Plausibility is 84.2% for *head_sanity* ($n = 38$) and 74.5% for *long_tail* ($n = 208$), pooling to 76.0% plausible and 85.8% training-relevant across the 246 judged edges.

Among the 59 IMPLAUSIBLE judgments, the dominant failure is *wrong_normalization_target* (54.2%), followed by *acceptable_but_weak* (20.3%) and *over_narrow_skill* (18.6%). It reflects the 0.7 title cosine threshold (Appendix D).