Overpredicting on Noise: Deployment-Faithful Evaluation of Skill Extraction Pipelines

Aleksander Bielinski[1,†], Warre Veys[2,3] and Jens-Joris Decorte[2]

[1]Edinburgh Napier University, School of Computing, Engineering & The Built Environment
[2]TechWolf, 9000 Gent, Belgium
[3]Ghent University – imec, 9052 Gent, Belgium

Abstract

The accelerating scale and volatility of the labor market make tracking skill demand an increasingly difficult and critical task. Skill extraction addresses this by mapping free-text job postings to a standardized taxonomy, yet existing work evaluates only on sentences already known to express a skill. This ignores the dominant case: a large fraction of job-description sentences express no skill at all (around 64% in the SKILL-XL dataset). As a result, systems are never penalized for surfacing skills on irrelevant text, which inflates reported performance and wastes computation on fragments that should be discarded. We introduce a first step towards a deployment-faithful evaluation protocol that includes no-skill sentences at their natural prior and rewards correct silence, and use it to show that current SOTA extractors, evaluated only on skill-bearing text, surface spurious skills on noise. We showcase that a lightweight relevance gate already substantially improves deployment-relevant ranking performance (MAP@5 $0.28 \rightarrow 0.66$) while passing roughly half as many sentences to the costly retrieval and ranking stages, thus halving the downstream computation. This demonstrates a benefit that skill-bearing-only evaluation is structurally unable to reveal, and opens up a new highly relevant avenue of research in this domain.

Keywords

Skill Extraction, Deployment-Faithful Evaluation, ESCO, Relevance Filtering, LLMs

1. Introduction

Fueled by technological developments and societal changes, today's labor market is transforming rapidly, making the assessment of skill demand an increasingly challenging task [1]. Understanding which competencies employers require, at scale and in near real time, is central to workforce planning, curriculum design, and labor-market policy. Skill extraction addresses this need by identifying competencies expressed in free-text sources such as job postings and resumes, and mapping them to a standardized taxonomy. The European Skills, Competences, Qualifications and Occupations (ESCO) taxonomy alone defines nearly 14,000 distinct skills, while job postings are published continuously at scale across online platforms. Reliably grounding this volume of noisy, unstructured text in a controlled vocabulary is therefore both a practical necessity and an open technical challenge, as reflected in a recent surge of research on efficient automated skill extraction [2, 3].

Research so far has focused on the specific aspects of the task. From binary classification, through span-level skill extraction, to finally sentence-to-skill retrieval, skill extraction has been studied extensively. However, there is currently no approach that considers the full input spectrum of the challenge. More recently, methods mapping job descriptions directly to standardized taxonomy entries have gained popularity, showing that modern transformer architectures retrieve relevant candidate skills effectively and that Large Language Models (LLMs) can rank them without fine-tuning. Yet across this line of work, the way these systems are evaluated shares a common blind spot. Notably, performance is measured only on sentences already known to express a skill. This ignores that a high proportion of a job description is noise, so systems are never penalized for surfacing skills on non-skill-bearing fragments. While this poses no issue in a test setting, it is misaligned with the real-world deployment scenario, where predicting skills on irrelevant text directly undermines the reliability of the system.

This need for deployment realism takes on added weight as the community converges on a shared, standardized basis for evaluation. The recent WorkRB framework [4], the first publicly available, community-driven benchmark suite for the work-research domain, is a clear marker of this shift, consolidating a broad range of tasks, ontologies, and models into a single, comparable framework. Yet the value of such a shared standard depends entirely on what it measures: once a community coordinates around a common benchmark, that benchmark defines what progress means, and any mismatch with real-world deployment is inherited by everyone who adopts it. This makes it all the more important that standardized evaluation reflects deployment conditions rather than performance on curated, skill-bearing text alone. Our work exposes precisely this gap, between how skill extraction systems are evaluated and how they must behave once deployed, and takes a concrete step towards closing it.

Recently, skill extraction has been increasingly approached as a retrieve-and-rank problem, where an initial list of candidate skills is refined by a dedicated ranking solution [5, 6]. This design is a natural fit for skill extraction, where relevant entries must be recovered from a taxonomy of thousands of labels [7]. Large language models are increasingly adopted for this second stage, acting as effective rankers without task-specific fine-tuning [8, 5]. Yet this line of work omits the dominant case, as a high proportion of sentences in a job description express no skill at all [9]. The consequence is concrete: given the boilerplate line *"Reasonable accommodations may be made to enable individuals with disabilities to perform the essential functions"*, a state-of-the-art retrieve-and-rank pipeline returns ten ESCO disability-care skills (*assist clients with special needs*,

RecSys in HR '26: The 6th Workshop on Recommender Systems for Human Resources, in conjunction with the 20th ACM Conference on Recommender Systems, September 28–October 2, 2026, Minneapolis, MN, United States.

† Corresponding author.
a.bielinski@napier.ac.uk (A. Bielinski); warre.veys@techwolf.ai (W. Veys); jensjoris@techwolf.ai (J. Decorte)
0000-0002-7254-1290 (A. Bielinski); 0009-0000-6838-9954 (W. Veys); 0009-0000-7892-1211 (J. Decorte)

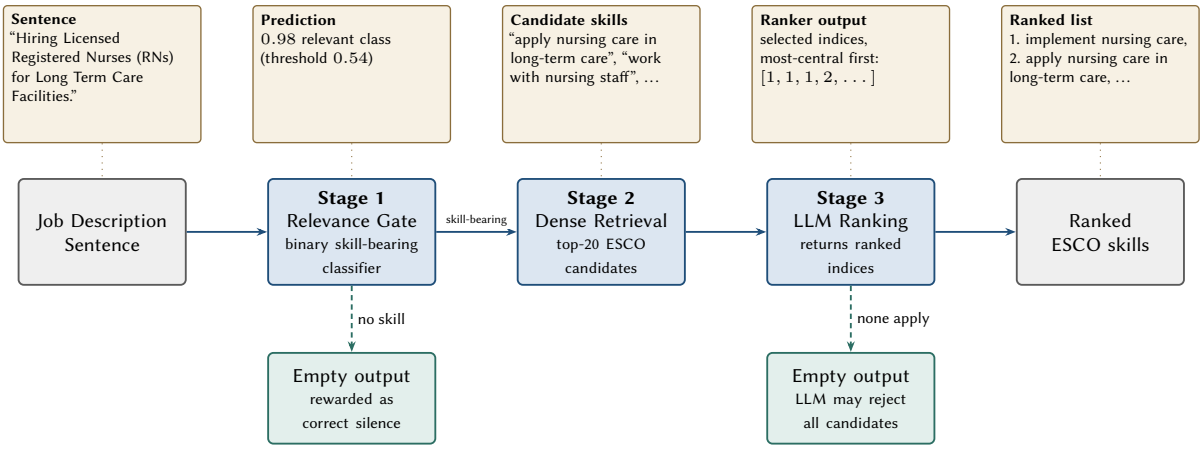

Figure 1: The three-stage skill extraction pipeline, evaluated under our deployment-faithful protocol. The relevance gate routes non-skill-bearing sentences to an empty output (rewarded as correct silence), while skill-bearing sentences proceed to retrieval and ranking. Tinted boxes trace one worked example through the pipeline.

support employability of people with disabilities, …), none of which the sentence expresses. Skill-bearing-only evaluation never scores this sentence, so the failure is invisible, yet a deployed system writes all ten into the vacancy's skill profile. As a result, reported performance can overstate real-world utility and hide the computational cost of ranking fragments that could and should have been discarded.

To address this, we introduce a first step towards a deployment-faithful evaluation protocol that includes non-skill-bearing sentences at their natural prior and rewards a system for correctly staying silent on them. To assess the protocol, we follow a modular three-stage pipeline (relevance gate, dense retrieval, and LLM ranking) that isolates the effect of each stage and reveals whether filtering noise before retrieval and ranking is necessary at all. Retrieval is performed against the ESCO taxonomy, aligning with the evaluation data and with state-of-the-art research in the domain. The protocol and pipeline are illustrated in Figure 1.

Using this protocol, we show that strong retrieve-and-rank systems, evaluated in the standard way, systematically over-predict on noise, even when calibrated with a threshold. Skill-bearing-only evaluation renders this failure invisible. A lightweight relevance gate placed before retrieval and ranking closes this gap, boosting deployment MAP@5 by 0.38 over the ungated baseline while passing roughly half as many sentences to the costly downstream stages. Notably, this simple gating step contributes more than any few-shot tuning of the ranker, underscoring that sentence extraction cannot be treated in isolation from the decision of whether a sentence warrants extraction at all.

Contributions. In summary, our main contributions are:

- A deployment-faithful evaluation protocol for skill extraction that scores systems under the natural class prior and rewards correct silence, exposing over-prediction on noise that skill-bearing-only evaluation cannot reveal.
- Evidence that strong retrieve-and-rank systems break down on noise, and that a lightweight relevance gate closes the gap (deployment MAP@5 $0.28 \rightarrow 0.66$) while passing roughly half as many sentences downstream, consistently across both retrievers.
- A publicly released binary relevance classifier, trained on a proprietary corpus enriched with public data, which enables this gating and generalizes across several held-out datasets.

2. Related Work

Early skill extraction was framed as span-level sequence labeling, training NER models to identify competency mentions in job postings and resumes [10, 11], later extended with generative LLMs [12, 13] and graph-based methods [14, 15]. A recurring limitation of span-based approaches is the lack of normalization: extracted spans are not linked to a standardized taxonomy, limiting real-world applicability [16, 17]. To ground extraction in a controlled vocabulary, subsequent work exploited large taxonomies such as ESCO, using skill labels and definitions as weak or distant supervision [18, 19, 20]. The high cost of expert annotation further motivated synthetic data generation, where taxonomy definitions improve example quality and can serve as a training signal in their own right [2, 3].

Methodologically, recent work converges on the retrieve-and-rank paradigm where a dense retriever proposes candidate skills from the taxonomy and a second stage refines their order [6]. Large language models are increasingly adopted for this second stage, acting as effective rankers without task-specific fine-tuning [8, 5, 21]. Crucially, across this body of research, both training and evaluation focus on skill-bearing sentences. The retriever is trained only on positive extraction, learning to rank a sentence's gold skills highly, and the accept/reject decision is deferred to a calibrated similarity threshold at inference [7]. Because the model is never exposed to non-skill-bearing sentences during training, this calibration is a fragile substitute for learning to abstain, as a threshold cannot reliably reject a noise sentence whose text sits close to a real skill in the embedding space. This overlooks that a significant portion of sentences in a job description express no skill at all [9], and measures ranking quality in isolation from the prior decision of whether a sentence warrants any prediction. While some prior studies address this by introducing explicit binary relevance filtering for retrieved skills (relevant

vs. not relevant) [3, 7, 6], they still evaluate filtering and ranking in isolation, never scoring the end-to-end system under the natural prevalence of non-skill-bearing text. Our work unifies both concerns in a single deployment-faithful setting that jointly rewards correct identification of skill-bearing sentences and correct ranking of the competencies they contain.

3. Deployment-Faithful Evaluation

Standard skill extraction metrics are computed only over sentences already known to express a skill. In an end-to-end setting, this masks a critical failure mode. A system that ranks relevant skills well, yet also emits skills for noise (e.g., company descriptions), produces a final list that misrepresents the posting's true requirements. Since a big portion of sentences in a job description carry no skill, a deployable system must not only rank correctly, but also stay silent when nothing should be extracted.

We therefore evaluate ranking under two modes. In the *Standard* mode, metrics follow their usual definition over skill-bearing sentences only, measuring pure ranking quality. In the *Deploy* mode, no-skill sentences are included at their natural prior and scored on correct silence. The gap between the two exposes over-prediction that skill-bearing-only evaluation cannot reveal.

4. Ranking Task and Data

Ideally, end-to-end evaluation would take a full job posting as input and return a single skill list ranked by relevancy. As no dataset of this form (posting → ranked skills) exists yet, we frame skill extraction as a sentence-level problem. Given a single sentence from a job description, the goal is to identify which ESCO skills the sentence genuinely expresses or requires, returned as a ranked list with the most clearly expressed skills first. Our evaluation data includes both skill-bearing and non-skill-bearing fragments, making it the closest available approximation of the real-world task. It preserves the two decisions a full-posting system must make: whether a fragment expresses any skill, and how to rank those it surfaces. Keeping the unit of prediction this small allows skills to be attributed precisely, while still reflecting how skill requirements are typically phrased in practice.

We evaluate on three benchmarks (see Table 1). SKILL-XL [7], a benchmark of job-posting sentences annotated with ESCO skills (v1.1.0, 13,891 distinct skill labels), provides the necessary annotations as distributed. TECH and HOUSE [19] do not: every one of their 517 and 403 sentences carries an annotated span, and a gate cannot be measured on a corpus with no negatives. However, both are ESCO relabellings of SKILLSPAN [10], whose public release retains every sentence of each posting, including the sentences annotators marked no span in, so the negatives were collected and adjudicated by the original annotators and simply dropped from the relabelled subset. We recover them by joining each benchmark sentence to SKILLSPAN on whitespace-normalized text, an exact join matching 517/517 TECH and 402/403 HOUSE sentences and recovering 43 and 34 source postings, then re-emitting those postings whole so that document context survives. A sentence carrying no SKILLSPAN span becomes a negative (2,133 TECH, 856 HOUSE). Setting aside the placeholder-label sentences discussed below (112 and 80) leaves 405 and 322 distinct sentences with a real ESCO label. A sentence occurring in several postings, being one unique evaluation sample in the original TECH/HOUSE datasets, is now a separate evaluation item in each posting, since the document context a gate potentially sees differs, which is why the skill-bearing counts in Table 1 exceed the number of distinct benchmark sentences (405 → 476 for TECH, 322 → 323 for HOUSE). Two further groups are removed from the re-emitted postings, and both sit outside every count quoted above: they are neither benchmark sentences nor recovered negatives. The first is sentences SKILLSPAN annotated a span in that never made it into the TECH/HOUSE release, so no ESCO label exists and their status is undetermined (115 and 10 occurrences). The second is exact repeats of a sentence within one posting (72 and 4), often noise or parsing artefacts, which would otherwise double-count in every metric. Neither is subtracted from the 517 and 403 benchmark sentences, from the skill-bearing totals, or from the negatives; removing them only shrinks the pool of recovered posting sentences.

We treat a sentence as skill-bearing only if it carries at least one assigned ESCO skill label. Ranking is evaluated with binary-relevance ranking metrics (Mean Average Precision), alongside set-based precision and recall. Table 2 shows a representative skill-bearing sentence with its gold skills and cluster ids, which group near-duplicate skills in the SKILL-XL dataset.

Table 1
Composition of the three benchmarks. We treat a sentence as skill-bearing if it carries ≥ 1 ESCO skill label. For SKILL-XL, relevant-but-unmapped sentences are treated as negatives. For TECH and HOUSE, recovered negatives are sentences from the same postings that SKILLSPAN annotators marked no span in, and counts exclude the placeholder-label sentences discussed below.

	SKILL-XL		TECH	HOUSE
	Test	Val.		
Skill-bearing (≥ 1 skill)	944	799	476	323
Relevant, no mapped skill	272	263	n/a	n/a
Not relevant, no skill	1,425	1,219	2,133	856
Total unique sentences	2,641	2,281	2,609	1,179
% skill-bearing	35.7	35.0	18.2	27.4

On sentences a human marked relevant that carry no ESCO label. All three benchmarks contain these: 272 test sentences in SKILL-XL ($\approx 10\%$), and 112 distinct TECH and 80 distinct HOUSE sentences whose annotated span was left with the placeholder LABEL NOT PRESENT or UNDERSPECIFIED. In each case an annotator judged the sentence relevant, but the dataset supplies no ESCO annotation for it, so we audited all of them with an LLM-assisted pass, to get an approximate judgement based on each sample's top-25 retrieved ESCO candidates.

In SKILL-XL only 6% admit a defensible label the annotator missed. Roughly half (52%) carry no skill content at all, being section headers, navigation fragments, marketing copy, or legal boilerplate, a further 17% are borderline, expressing only vague traits, and around a quarter (26%) state a genuine requirement that ESCO cannot express, chiefly certifications, education levels, and some physical demands (*"Standing", "Use of Hands"*). Arguing that these are also irrelevant when working with a vocabulary that does not

represent close skills, we treat these sentences as negatives: the system must produce no output for them.

In TECH and HOUSE, the same audit finds a correct ESCO label in half (50%) of the cases, so scoring them as negatives would misscore half the population. Restoring them as positives is not available to us either, because those labels come from the pipeline's own retriever (96% fall inside the top-20 candidates the ranker is handed) rather than from a human, and grading the system against its own suggestions would flatter it. We therefore exclude them all, 206 sentence occurrences across the re-emitted postings, at a cost of 4.6% of TECH and 6.4% of HOUSE, which keeps every gold label human-assigned.

Table 2
A skill-bearing sentence from SKILL-XL with its gold ESCO skills and cluster ids. Clusters group near-duplicate skills (e.g., *work in teams* and *teamwork principles*).

"As a Customer Sales Guide, you will use a team-based approach to ensure a one-of-a-kind sales experience for our customers, helping them navigate the path to vehicle ownership at the dealership."

Gold skill	Cluster (near-duplicate group)
work in teams	1
teamwork principles	1
manage the customer experience	2
sell vehicles	3
advise customers on motor vehicles	3

The test splits are reserved for evaluation, whereas the validation split of SKILL-XL is used to mine the few-shot demonstrations for the ranking stage (see Section 5.3).

5. Pipeline

This section describes the proposed end-to-end skill extraction pipeline. It covers the whole process from sentence relevance classification (5.1), through relevant candidate sourcing (5.2) to final skill ranking (5.3). The pipeline design allows for swapping models at each stage, thus allowing for the incorporation of novel approaches as methods improve.

5.1. Relevance Gate

The purpose of the binary classifier is to identify skill-bearing sentences suitable for retrieval and ranking. Motivated by the noisy nature of the job description data, such a gating mechanism translates to faster processing and narrows down the task in later stages. Both of these are beneficial when it comes to real-world deployment settings, as retrieval, and especially ranking, are more computationally expensive and demanding tasks. In other words, the initial binary classification can be viewed as a cheap method for cleaning the dataset. Furthermore, as binary relevance requires less domain expertise than explicit skill classification, the training of such models is more feasible.

5.1.1. Model and Data

Due to the lack of open-source binary classifiers, in this work, a dedicated binary sentence classifier was developed by fine-tuning RoBERTa [22] with LoRA [23], with the resulting adapter merged into a standalone checkpoint [1]. Training data was drawn from two sources. The first is a proprietary corpus of job-posting sentences, each labeled for binary relevance by two trained annotators without domain expertise but with prior training provided. The data, retrieved using the Lightcast API[2], spans UK job postings from January 2019 to October 2022. The original pool contained 16,900 adverts ($\approx 250{,}000$ sentences), from which 10% of adverts ($\approx 25{,}000$ sentences) were sampled for annotation[3]; all identifiers were removed in compliance with Lightcast's terms of service. Inter-annotator agreement, measured on a doubly-annotated subset of 2,471 sentences, reaches a Cohen's κ of 0.69, indicating substantial consistency. The second enriches this corpus with three publicly available skill-extraction datasets, SKILLSPAN [10], SAYFULLINA [24], and GREEN [25], which are natively span-annotated rather than binary. We convert each to sentence-level relevance by marking any sentence containing at least one skill span as relevant; for GREEN, only skill spans are counted, as the dataset additionally annotates occupations and certifications that fall outside our target. Merging all training material yields a pool of 40,451 sentences (relevant-to-irrelevant ratio 1.73:1), from which 20% is held out for validation. At inference, the classifier scores each sentence and drops those below a decision threshold of 0.54, tuned on the validation split. Per-dataset composition alongside the annotation details can be found in Appendix B.

5.2. Dense Retrieval

The retriever compares each sentence against the full ESCO skill taxonomy and returns a candidate list of skills. All of the retrievers used in this study are bi-encoders, encoding queries (job-ad sentences) and documents (skills) independently, thus efficiently. However, other architectures may be substituted, provided they remain consistent with the target taxonomy and return a fixed list of candidate skills.

Each sentence is embedded and matched against all ESCO skill labels by cosine similarity, retaining the top $N = 20$ candidates. As the retriever does not guarantee that every gold skill appears among these candidates, downstream ranking quality is upper-bounded by retrieval recall. This is a genuine deployment constraint that our evaluation accounts for rather than hides. The sole exception is few-shot demonstration mining. Specifically, when preparing demonstrations for the LLM ranker, we inject any missed gold skills into the candidate pool, ensuring each demonstration shows a complete, correctly ranked answer. This injection is never applied at evaluation time.

5.3. LLM Ranking

Ranking is the most demanding task of the pipeline. Determining which candidate skills a sentence genuinely expresses, and ordering them by relevance, requires nuanced semantic judgment, and doing so consistently would traditionally demand a model trained on task-specific relevance annotations. Supervision at such a level is costly to obtain and scarce, particularly in a job domain. The task is further

[1] https://huggingface.co/Aleksandruz/Binary-Relevance-RoBERTa
[2] https://docs.lightcast.dev/
[3] Due to time constraints, the final annotated set comprised 21,198 sentences.

complicated by the need to decide not only how to order candidates, but whether any of them apply at all, since a retrieved candidate list says nothing about whether its sentence is genuinely skill-bearing. Large language models are well-suited to these joint demands. First, they perform nuanced relevance judgment without task-specific fine-tuning. Secondly, prompting strategies allow for both selection and ordering of candidates, as well as returning an empty answer when none apply. We exploit this by casting ranking as a constrained selection problem over the retrieved candidates, described below.

5.3.1. Prompting Strategy

At inference time, the LLM is provided with the specific system prompt and, if applicable, the selection of demonstrations (see Appendix A). While the entire purpose of the gating stage is to filter out all of the irrelevant job description sentences, the LLM must retain the option to treat all candidate skills as irrelevant. This was done to ensure a comparable baseline where no gating mechanism is used, but also to provide an additional correction mechanism for the gate's false positives. To ensure the LLM has exposure to both competence-bearing and irrelevant skill sentences, the demonstrations for these two categories are sampled independently. This allows us to assess the effect of providing the model with specific demonstrations while also helping to determine the optimal number of them.

The demonstrations are sourced from the gold-injected validation split of SKILL-XL. For each retriever variant, a separate demonstration pool is created, isolating the effect of each retriever and allowing for a more robust assessment of ranking performance. Rather than sampling demonstrations at random, we select them to maximize coverage of the required output structures, following evidence that diverse demonstrations improve in-context generalization [26]. For the skill-bearing demonstrations, we prioritize examples whose gold skills span the most distinct clusters; since clusters group near-duplicate skills, these are the demonstrations covering the most genuinely different skills. For the no-gold demonstrations, priority is given to sentences that were assigned a relevant label in the data but map to no skill (see Section 4), exposing the model to the hardest rejection cases.

6. Experimental Setup

All of the experiments are performed using an NVIDIA RTX 4070 Ti SUPER with 16 GB of VRAM. The repository containing all the details on how to implement the evaluation protocol can be found here: https://github.com/AleksanderB-hub/Ranking_Pipeline_Final.

6.1. Models

The binary relevance stage uses the classifier described in Section 5.1.1. We keep the decision threshold from its original training regime and, although a SKILL-XL validation split is available, deliberately avoid any further tuning, so that no pipeline component is adapted to the evaluation data. Full implementation details are given in Appendix C.

For retrieval, we use ConTeXTMatch [7] and CurriculumMatch [6], each loaded from its respective Hugging Face checkpoint.

For LLM ranking, we use Qwen3-14B-AWQ [27] served via vLLM [28], likewise from a Hugging Face checkpoint. The model provides a built-in "thinking" mode for multi-step reasoning; preliminary runs showed no benefit in our setting, so it was disabled throughout for efficiency. For full details, see Appendix C.

6.2. Evaluation Protocol

Our evaluation spans two main areas. First, the base binary relevance model performance is reported across a range of proprietary and publicly available data. This establishes its suitability for the task. The datasets used for this were described in more detail in 5.1.1.

When testing the proposed pipeline, we use the three benchmarks introduced in Section 4. Both the binary classification and candidate retrieval models, and their training data, are separate from these benchmarks. This was dictated by the need for a deployment-faithful evaluation protocol and offers a more robust assessment, since no pipeline component has been trained on the evaluation data. This reduces the risk that reported performance reflects in-distribution overfitting rather than genuine skill-extraction ability.

6.2.1. Metrics

For a binary gate, we report accuracy and F1 scores. Additionally, to investigate the nature of the failure cases, a classification matrix is provided for the SKILL-XL dataset. Retrieval is reported as a diagnostic rather than a headline result, using Recall@20 over skill-bearing sentences only; these bound downstream ranking, since a skill the retriever misses can never be ranked. Recall is undefined for no-skill sentences and is therefore not reported on them.

For ranking, we report Mean Average Precision (MAP) at $k \in \{5, 10\}$, under both the *Standard* and *Deploy* modes defined in Section 3. These cut-offs suit the dataset, where skill-bearing sentences average 4.73 gold skills and only 64 exceed ten. In the *Deploy* mode, no-skill sentences are included and scored on correct silence: a sentence receives 1.0 if the pipeline predicts nothing and 0.0 otherwise, penalizing spurious predictions on noise. Relevance is binary, matching the annotations: since SKILL-XL's cluster ids group near-duplicate skills rather than grading relevance, graded ranking metrics such as normalized Discounted Cumulative Gain would have no basis in the data, and we do not report them.

6.2.2. Ranking: Few-shot configuration

The number of few-shot demonstrations shown to the LLM ranker is a key variable, as the two demonstration types tune opposing behaviors: skill-bearing examples teach the model to produce and order candidates, while no-gold examples teach it to abstain. To characterize this trade-off, we sweep the number of skill-bearing demonstrations $n \in \{0, 1, 2, 3\}$ and no-gold demonstrations $m \in \{0, 1, 2, 3\}$, drawn from disjoint subsets of the demonstration pool (Section 5.3), and additionally evaluate the zero-shot configuration ($n = m = 0$). Each configuration is run under both the gated and ungated pipeline, allowing us to isolate the effect of demonstration composition from the effect of the relevance gate. All demonstrations are selected deterministically, so that differences across configurations reflect their composition rather than sampling variance.

7. Results

7.1. Relevance Classification

We first assess the relevance classifier as a standalone component. Table 3 reports accuracy and macro-F1 across five held-out test sets spanning proprietary and public data. The classifier performs strongly across all sets with a genuine class balance, reaching macro-F1 between 0.83 and 0.91, confirming that binary relevance transfers reliably across annotation schemes, including datasets natively labeled at the span level.

Table 3
Standalone binary relevance classifier performance across held-out test sets (tuned decision threshold).

Test set	Accuracy	Macro-F1
Proprietary (expert)	0.861	0.829
Proprietary (cross-annot)	0.871	0.860
GREEN	0.845	0.828
SKILLSPAN	0.922	0.905
SAYFULLINA*	0.994	0.498

*SAYFULLINA's test split contains only two negative sentences, so macro-F1 collapses despite near-perfect accuracy; it is reported for completeness only.

Crucially, on SKILL-XL, unseen during training, the gate still reaches an F1 of 0.828, demonstrating genuine out-of-distribution generalization, with a highly asymmetric error profile. False positives (337), as shown in Figure 2, far outnumber false negatives (39), making it a high-recall, lower-precision filter (R=0.959, P=0.729) that rarely discards a skill-bearing sentence but admits some irrelevant ones. As every false positive is a sentence on which the ranker may produce spurious skills, this residual imprecision is the gate's main limitation and propagates into the end-to-end results (Section 7.2). An optional LLM audit of the gate's uncertain band can push precision further, raising *Deploy* MAP@5 from 0.66 to 0.72 at a small cost to recall (see Appendix D).

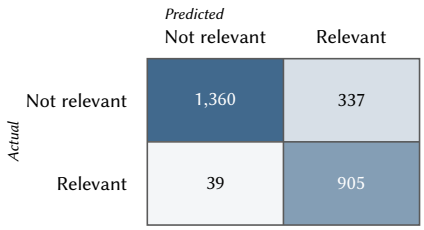

Figure 2: Relevance gate on the SKILL-XL test split. False positives (337) outnumber false negatives (39) by nearly an order of magnitude.

7.2. End-to-End: Gating Effect

Since ranking performance depends on candidate retrieval quality, we first report how well each retriever surfaces relevant skills (see Table 4, full results in Appendix E). Measured at the retrieval stage over skill-bearing sentences only (and thus independent of gating), CurriculumMatch retrieves 62.1% of gold skills within its top-20 list (R@20, based on cosine-similarity) which is the stronger of the two. This reveals the core bottleneck in the pipeline, as nearly 38% of gold skills are never surfaced to the ranker and are therefore unrecoverable downstream, regardless of ranking quality.

Considering the set-based metrics, precision rises substantially across all retrievers once relevance gating is applied. The skill-bearing $F1_{sb}$ barely changes while the overall $F1_{all}$ increases substantially: because the gate leaves predictions on skill-bearing sentences largely intact but suppresses spurious predictions on noise, the gains come almost entirely from filtering non-skill-bearing sentences rather than from better extraction on relevant ones.

At the natural class distribution, the gate rejects 53% of sentences (1,399 of 2,641), which are never passed to the costly retrieval and ranking stages. Each routed-away sentence saves a similarity search over all 13,891 ESCO labels and a full LLM ranking call. In deployment, these systems run over high volumes of postings, each yielding many sentences, so per-sentence savings accumulate quickly.

Retrieval recall, however, is only one side of the problem. Without gating, non-skill-bearing sentences are still processed, and the ranker readily assigns skills to them, as the boilerplate example in Section 1 illustrates. The *Deploy* setting quantifies this.

When it comes to ranking capability, in the *Standard* setting, relevance gating slightly reduces performance across all retrievers. This is expected as the setting scores only skill-bearing sentences, so the gate can only remove genuine gold sentences it wrongly rejects, never help. Extending the list from k=5 to k=10 yields little benefit and even lowers MAP, since most sentences carry no more than five skills.

The *Deploy* setting paints a completely different picture. By suppressing non-skill-bearing sentences, gating more than doubles deployment ranking quality across every retriever. For reference, an always-silent system scores 0.643 on Deploy MAP@5 under this prior, purely from correct abstention. The ungated pipeline (0.281) falls well below this floor, confirming that a naively deployed retrieve-and-rank system is worse than one that predicts nothing. Gating lifts it to 0.664, above the trivial baseline, while its Standard MAP and $F1_{all}$ (both 0 for the silent system) confirm it is genuinely ranking rather than merely abstaining.

Most of that movement is abstention rather than ranking, and it is worth reporting the two separately. The Silence column of Table 4 makes this explicit: on SKILL-XL the gated pipeline returns nothing for 81% of no-skill sentences, against 21–23% ungated, at the cost of a mute response on 4–5% of skill-bearing ones. This correct-silence rate is near-identical across retrievers on every benchmark (81.4–81.7% on SKILL-XL, 94.4–94.5% on TECH, 89.1–89.4% on HOUSE), confirming it is the gate and not the retriever that produces it, and it rises with the noise level of the benchmark. The ungated rate varies more with the retriever and the corpus (20–32%), but never approaches the gated regime. Read this way the margin over silence is also visibly narrow, between 0.017 and 0.045 across the three benchmarks: gating raises the floor reliably, but it does not transform ranking quality.

This gain reflects the pipeline correctly staying silent on noise. This is precisely the behavior a deployed system requires, and that standard ranking metrics never measure. The scale of this gain depends on the class prior, as a noisier dataset simply means more sentences for the gate to correctly silence. Nonetheless, the underlying per-class behavior remains the same regardless of the split. Unlike in the *Standard* setting, the threshold-tuned baseline (B) here

Table 4
End-to-end performance at the headline configuration ($n0{+}m1$: zero skill-bearing, one non-skill-bearing demonstration) across all three benchmarks. R@20 is the retrieval ceiling, measured over skill-bearing sentences independently of gating. $F1_{\text{all}}$ scores every sentence, $F1_{\text{sb}}$ only skill-bearing ones. Silence is the correct-silence rate: the percentage of non-skill-bearing sentences on which the system returns nothing. MAP@5 is the *Standard* (skill-bearing only) score and MAP_{dep} the *Deploy* score. (B) predicts skills directly from retriever cosine scores at a validation-tuned threshold ($\tau = 0.538$), without the LLM ranker. Best per column within each benchmark in bold.

Retriever	Pipeline	R@20	$F1_{\text{all}}$	$F1_{\text{sb}}$	avg_preds	Silence (%)	MAP@5	MAP_{dep}
SKILL-XL (2,641 sentences, 64% non-skill-bearing)								
Always-silent	n/a	n/a	0.000	0.000	0.00	100.0	0.000	0.643
CurriculumMatch (B)	τ	0.621	0.165	0.219	1.56	62.1	0.209	0.474
CurriculumMatch	ungated	0.621	0.160	**0.300**	6.91	20.9	**0.413**	0.282
	gated		**0.249**	0.298	3.63	81.4	0.395	**0.664**
ConTeXTMatch	ungated	0.572	0.145	0.271	6.79	22.5	0.393	0.285
	gated		0.226	0.270	3.56	**81.7**	0.378	0.660
TECH (2,609 sentences, 82% non-skill-bearing)								
Always-silent	n/a	n/a	0.000	0.000	0.00	100.0	0.000	0.818
CurriculumMatch (B)	τ	0.805	0.109	0.248	1.48	58.9	0.333	0.542
CurriculumMatch	ungated	0.805	0.059	0.211	6.62	30.6	**0.543**	0.349
	gated		**0.175**	0.214	1.87	94.4	0.501	**0.863**
ConTeXTMatch	ungated	0.748	0.054	0.190	6.55	31.5	0.513	0.351
	gated		0.155	0.189	1.91	**94.5**	0.462	0.857
HOUSE (1,179 sentences, 73% non-skill-bearing)								
Always-silent	n/a	n/a	0.000	0.000	0.00	100.0	0.000	0.726
CurriculumMatch (B)	τ	0.720	0.126	0.197	1.08	68.3	0.187	0.547
CurriculumMatch	ungated	0.720	0.090	**0.210**	6.27	20.3	**0.464**	0.275
	gated		**0.171**	0.207	2.89	89.1	0.433	**0.766**
ConTeXTMatch	ungated	0.662	0.081	0.191	6.31	21.7	0.428	0.275
	gated		0.154	0.190	2.89	**89.4**	0.394	0.757

Table 5
Relevance gate transferred to the reconstructed benchmarks, unseen during its training. Reported over all sentences at the deployed threshold.

	SKILL-XL	TECH	HOUSE
Precision	0.729	0.777	0.762
Recall	0.959	0.914	0.944
F1	0.828	0.840	0.844
Accuracy	0.858	0.936	0.904

outperforms the ungated LLM ranker (Deploy MAP@5 0.474 vs. 0.281), since its conservative threshold suppresses predictions on many noise sentences. Even so, it trails the gated pipeline (0.664), confirming that simple threshold tuning remains suboptimal for ranked skill extraction.

7.3. Few-Shot Composition

Having established the efficacy of the gated pipeline at the headline $n0{+}m1$ configuration, we examine how demonstration composition shapes downstream behavior. We vary the number of skill-bearing (n) and non-skill-bearing (m) demonstrations, focusing on extraction quality ($F1_{all}$), ranking performance (MAP@5), and verbosity (avg_preds). We report the ungated CurriculumMatch pipeline, where the demonstrations act on the LLM directly without the gate masking their effect. The full breakdown across retrievers and the gated setting is given in Appendix E.

Results in Table 6 reveal that the two demonstration types tune opposing behaviors. Skill-bearing demonstrations improve ranking, raising the *Standard* MAP@5 from 0.408 (zero-shot) to 0.432 ($n3{+}m0$), but drive verbosity up sharply (7.2 → 13.7 predictions) as the model treats more candidates as valid. Non-skill-bearing demonstrations do the opposite: a single rejection example ($n0{+}m1$) tightens output to 6.9 predictions while lifting $F1_{all}$ to 0.160, near the peak of 0.162 at $n0{+}m3$ configuration. The best composition therefore depends on the downstream objective, with ranking demonstrations favoring ordering quality and rejection demonstrations targeting precision. Although our objective prioritizes ranking, we select $n0{+}m1$ as the headline configuration. Its ranking quality is only marginally below the best ($0.432 \rightarrow 0.412$), yet it almost perfectly matches the strongest $F1_{all}$ while using a single demonstration, keeping prompts short and inference fast.

8. Discussion

The results showed that a simple gating mechanism can substantially improve downstream skill extraction performance. Specifically, under deployment-faithful evaluation, systems must both rank competences with respect to the job description fragment and discard non-competence-bearing queries. This is why we introduce the *Deploy* setting, a step towards a more demanding evaluation standard. Given that much of a job description is noise [9], such gating mechanisms are a cheap addition that secures a meaningful improvement. Crucially, this improvement is not only in quality but also in cost: by discarding noise before the expensive retrieval

Table 6
Effect of demonstration composition on the ungated CurriculumMatch pipeline for different skill-bearing (n) and non-skill-bearing demonstrations (m) configurations. Best per column in bold.

Config	$F1_{\text{all}}$	MAP@5	avg_preds
Zero-shot	0.150	0.4076	7.19
n0+m1	0.160	0.4122	**6.91**
n0+m3	**0.162**	0.4174	7.15
n1+m0	0.113	0.4224	13.07
n1+m1	0.136	0.4195	9.92
n2+m3	0.128	0.4292	11.02
n3+m0	0.110	**0.4317**	13.66
n3+m2	0.125	0.4316	11.74
n3+m3	0.121	0.4312	12.03

and ranking stages, the gate reduces per-sentence inference at a scale that matters for high-volume deployment.

The demonstration analysis reveals that simplicity outweighs few-shot tuning: a cheap binary classifier moves deployment performance more than any demonstration composition. Prior work on LLMs for skill ranking either ignored the relative ordering of skills or treated the task as pure retrieval, with no penalty for surfacing non-skill-bearing entries [8, 5]. The LLM ranker is by no means perfect, as the moderate *Standard* ranking performance suggests, and it is also constrained by retrieval quality: if the ranker does not see a sentence's full set of candidate skills, it cannot rank them perfectly. Given the data scarcity in the domain, the out-of-the-box usage and fast inference secured by the optimal prompting configuration make LLM rankers particularly well-suited for the task.

Furthermore, the lack of data dictates that existing resources be used strategically, improving downstream extraction without heavy reliance on new annotation. Although our binary classifier is trained partly on proprietary data, it is the pipeline structure and not the training data that drives the gains, which is why we release the model and code so the approach can be reused without that data. We hope this encourages end-to-end skill-extraction assessment, making such systems more robust and trusted for stakeholders such as policymakers and HR professionals, and supporting wider adoption and richer data over time.

8.1. Limitations and Future Work

A main limitation is the retrieval ceiling, which at best leaves 38% of gold skills unrecoverable regardless of ranking. This is largely consistent with previous work on skill retrievers [6], but remains a significant bottleneck, so further work should improve candidate retrieval, especially for the rarer skills that are hardest to retrieve.

A second limitation arises at the gate. Because it only suppresses sentences, its false rejections (4% of skill-bearing sentences on SKILL-XL, 6% on HOUSE, 9% on TECH) are unrecoverable downstream: the gate improves deployment-faithful performance precisely by accepting a loss of relevant sentences. A confidence-aware gate that defers uncertain sentences could soften this.

We evaluate on three benchmarks: SKILL-XL, and the reconstructed TECH and HOUSE (Section 4). The latter two share a source corpus and annotation protocol, so they are not fully independent, and their postings come from a single technical job platform and one in-house collection. Broader evidence would require different corpora and annotation schemes, so our pipeline, and specifically the binary relevance classifier, should be revisited as new datasets emerge. Our findings likewise rest on a single ranking model (Qwen3-14B-AWQ), selected for realistic local-deployment constraints rather than as a frontier system.

Finally, all three benchmarks contain skills ESCO cannot express: a quarter of SKILL-XL's relevant-but-unmapped sentences state a requirement with no taxonomy entry, as do 51 of the 192 distinct placeholder-label sentences in TECH and HOUSE, mostly named technologies (`Docker`, `Kubernetes`, `Kafka`). Our protocol scores silence on these as correct, rewarding a system for withholding a real but unstandardised competence. Taxonomy-relative evaluation thus inherits the taxonomy's blind spots, and unevenly: emerging or niche skills stay invisible until ESCO catches up, which matters when such systems inform labour-market analytics.

9. Conclusion

In this work, we argued that skill extraction should be evaluated in its real-world deployment context, where a high proportion of sentences carry no skill and a reliable system must stay silent on them, which current work overlooks by evaluating only on skill-bearing text. To address it, we introduced a deployment-faithful evaluation protocol that includes no-skill sentences at their natural prior and rewards correct silence, and instantiated a modular three-stage pipeline (relevance gate $\rightarrow$ retrieval $\rightarrow$ LLM ranking).

Our results showed that a lightweight relevance gate, placed before the costly retrieval and ranking stages, substantially improves deployment performance (MAP@5 $0.28 \rightarrow 0.66$) while roughly halving the sentences passed downstream. This gain is invisible to standard skill-bearing evaluation, holds consistently across all retrievers tested, and moves the needle more than any few-shot demonstration composition. We release our binary relevance classifier and all code to support further end-to-end assessment, and hope this encourages the community to evaluate skill extraction under conditions that reflect real-world deployment.

Ethical Considerations

We evaluate on publicly available datasets, and we do not process personal data. The data used to train the binary classifier cannot be released due to licensing agreements. This publication uses the ESCO classification of the European Commission. Job ads and standardized taxonomies may encode societal and market biases (e.g., occupational stereotypes), so our system is intended for labor-market analytics and skill insights rather than automated hiring decisions, and we recommend human oversight for any downstream HR use. We release code and configuration files for reproducibility.

Declaration on Generative AI

The authors used *Grammarly* and *Claude.ai* for spelling, grammar, and paraphrasing, then reviewed and edited all resulting content and take full responsibility for it.

References

[1] W. E. Forum, The Future of Jobs Report 2025, Technical Report, Amherst, MA, USA, 2025.

[2] Decorte, Jens-Joris and Verlinden, Severine and Van Hautte, Jeroen and Deleu, Johannes and Develder, Chris and Demeester, Thomas, Extreme multi-label skill extraction training using large language models, in: AI4HR & PES, ECML-PKDD 2023 Workshop, Proceedings, 2023, pp. 1–10.

[3] A. Magron, A. Dai, M. Zhang, S. Montariol, A. Bosselut, JobSkape: A framework for generating synthetic job postings to enhance skill matching, in: E. Hruschka, T. Lake, N. Otani, T. Mitchell (Eds.), Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024), Association for Computational Linguistics, St. Julian's, Malta, 2024, pp. 43–58. URL: https://aclanthology.org/2024.nlp4hr-1.4/.

[4] M. D. Lange, W. Veys, F. Retyk, D. Deniz, W. Jouanneau, M. Zhang, A. Bielinski, E. Jouffroy, N. Clobes, N. Baranowska, D. Graus, M. Palyart, R. Zbib, D. Gkatzia, T. Demeester, T. D. Bie, T. Bogers, J.-J. Decorte, J. V. Hautte, Workrb: A community-driven evaluation framework for ai in the work domain, 2026. URL: https://arxiv.org/abs/2604.13055. arXiv:2604.13055.

[5] K. D'Oosterlinck, O. Khattab, F. Remy, T. Demeester, C. Develder, C. Potts, In-context learning for extreme multi-label classification, arXiv preprint arXiv:2401.12178 (2024).

[6] A. Bielinski, D. Brazier, From Retrieval to Ranking: A Two-Stage Neural Framework for Automated Skill Extraction, in: Bogers, Toine and Bied, Guillaume and Decorte, Jean-Joris and Johnson, Chris and Kaya, Mesut (Ed.), Proceedings of the 5th Workshop on Recommender Systems for Human Resources (RecSys-in-HR 2025), in conjunction with the 19th ACM Conference on Recommender Systems, volume 4046, CEUR, 2025, p. 10. URL: https://ceur-ws.org/Vol-4046/RecSysHR2025-paper_5.pdf.

[7] Decorte, Jens-Joris and Van Hautte, Jeroen and Develder, Chris and Demeester, Thomas, Efficient text encoders for labor market analysis, IEEE ACCESS 13 (2025) 133596–133608. URL: http://doi.org/10.1109/ACCESS.2025.3589147.

[8] B. Clavié, G. Soulié, Large language models as batteries-included zero-shot esco skills matchers, in: Proceedings of the 3rd Workshop on Recommender Systems for Human Resources (RecSys in HR 2023), in conjunction with the 16th ACM Conference on Recommender Systems, Association for Computing Machinery, Singapore, 2023.

[9] I. Khaouja, G. Mezzour, I. Kassou, Unsupervised skill identification from job ads, in: 2021 IEEE 22nd International Conference on Information Reuse and Integration for Data Science (IRI), 2021, pp. 147–151. doi:10.1109/IRI51335.2021.00026.

[10] M. Zhang, K. Jensen, S. Sonniks, B. Plank, SkillSpan: Hard and soft skill extraction from English job postings, in: M. Carpuat, M.-C. de Marneffe, I. V. Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 4962–4984. URL: https://aclanthology.org/2022.naacl-main.366. doi:10.18653/v1/2022.naacl-main.366.

[11] M. Zhang, R. van der Goot, M.-Y. Kan, B. Plank, NNOSE: Nearest neighbor occupational skill extraction, in: Y. Graham, M. Purver (Eds.), Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, St. Julian's, Malta, 2024, pp. 589–608. URL: https://aclanthology.org/2024.eacl-long.35.

[12] K. Nguyen, M. Zhang, S. Montariol, A. Bosselut, Rethinking skill extraction in the job market domain using large language models, in: E. Hruschka, T. Lake, N. Otani, T. Mitchell (Eds.), Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024), Association for Computational Linguistics, St. Julian's, Malta, 2024, pp. 27–42. URL: https://aclanthology.org/2024.nlp4hr-1.3.

[13] A. Herandi, Y. Li, Z. Liu, X. Hu, X. Cai, Skill-llm: Repurposing general-purpose llms for skill extraction, arXiv preprint arXiv:2410.12052 (2024).

[14] I. Konstantinidis, M. Maragoudakis, I. Magnisalis, C. Berberidis, V. Peristeras, Knowledge-driven unsupervised skills extraction for graph-based talent matching, in: Proceedings of the 12th Hellenic Conference on Artificial Intelligence, 2022, pp. 1–7.

[15] N. Goyal, J. Kalra, C. Sharma, R. Mutharaju, N. Sachdeva, P. Kumaraguru, Jobxmlc: Extreme multi-label classification of job skills with graph neural networks, in: Findings of the Association for Computational Linguistics: EACL 2023, 2023, pp. 2181–2191.

[16] A.-s. Gnehm, E. Bühlmann, H. Buchs, S. Clematide, Fine-grained extraction and classification of skill requirements in German-speaking job ads, in: D. Bamman, D. Hovy, D. Jurgens, K. Keith, B. O'Connor, S. Volkova (Eds.), Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS), Association for Computational Linguistics, Abu Dhabi, UAE, 2022, pp. 14–24. URL: https://aclanthology.org/2022.nlpcss-1.2/. doi:10.18653/v1/2022.nlpcss-1.2.

[17] D. C. Kavargyris, K. Georgiou, E. Papaioannou, K. Petrakis, N. Mittas, L. Angelis, Escox: A tool for skill and occupation extraction using llms from unstructured text, Software Impacts (2025) 100772.

[18] M. Zhang, K. N. Jensen, R. van der Goot, B. Plank, Skill extraction from job postings using weak supervision, in: Proceedings of RecSys in HR'22: The 2nd Workshop on Recommender Systems for Human Resources, in conjunction with the 16th ACM Conference on Recommender Systems, Association for Computing Machinery, Seattle, 2022.

[19] J.-J. Decorte, J. Van Hautte, J. Deleu, C. Develder, T. Demeester, Design of negative sampling strategies for distantly supervised skill extraction, in: Kaya, Mesut and Bogers, Toine and Graus, David and Mesbah, Sepideh and Johnson, Chris and Gutiérrez, Francisco (Ed.), Proceedings of the 2nd Workshop on Recommender Systems for Human Resources (RecSys-in-HR 2022), volume 3218, CEUR, 2022, p. 7. URL: https://ceur-ws.org/Vol-3218/RecSysHR2022-paper_4.pdf.

[20] M. Zhang, R. van der Goot, B. Plank, ESCOXLM-R: Multilingual taxonomy-driven pre-training for the job market domain, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguis-

tics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 11871–11890. URL: https://aclanthology.org/2023.acl-long.662/. doi:10.18653/v1/2023.acl-long.662.

[21] E. A. İltüzer, Ö. A. Özlü, V. Farajijobehdar, G. Eryiğit, Leveraging llms for turkish skill extraction, arXiv preprint arXiv:2601.22885 (2026).

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[23] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al., Lora: Low-rank adaptation of large language models., Iclr 1 (2022) 3.

[24] L. Sayfullina, E. Malmi, J. Kannala, Learning representations for soft skill matching, in: Analysis of Images, Social Networks and Texts: 7th International Conference, AIST 2018, Moscow, Russia, July 5–7, 2018, Revised Selected Papers 7, Springer, 2018, pp. 141–152.

[25] T. A. Green, D. Maynard, C. Lin, Development of a benchmark corpus to support entity recognition in job descriptions, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, 2022, pp. 1201–1208.

[26] I. Levy, B. Bogin, J. Berant, Diverse demonstrations improve in-context compositional generalization, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2023, pp. 1401–1422. URL: https://aclanthology.org/2023.acl-long.78/. doi:10.18653/v1/2023.acl-long.78.

[27] Q. Team, Qwen3 technical report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

[28] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, I. Stoica, Efficient memory management for large language model serving with pagedattention, in: Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles, 2023.

[29] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2623–2631.

A. LLM Ranking Configuration

The prompt templates used for the LLM ranking both in zero-shot and few-shot variants. **System Prompt:** *You identify which skills a job-description sentence expresses, and rank them by how central they are to the sentence. You are given one sentence and a numbered list of candidate ESCO skills retrieved for it. Return the numbers of the candidates that the sentence genuinely expresses or requires, ordered from most to least central, the skill the sentence is most clearly about first. Select a candidate only if the sentence clearly supports it; do not guess. If the sentence expresses none of the candidates, or expresses no skill at all, return an empty list. Respond with only a JSON object of the form* `{"ranked": [<numbers>]}` *and nothing else, ordered most relevant first.*

B. Binary Relevance Classification

Table 7 reports the binary relevance classification model's training details. The Proprietary "expert" set represents the examples annotated by the labor market domain expert, with "cross-annot" being the cross-annotated (2 annotators) portion of the dataset. The details on the model and annotation can be found here: https://github.com/AleksanderB-hub/Job_Relevance_Classification.

Table 7
Composition of the binary relevance training and evaluation data. Public datasets (SKILLSPAN, SAYFULLINA, GREEN) are converted from span-level to binary relevance.

Split	Source	Relevant	Irrelevant
Train	Proprietary	11860	5424
	SKILLSPAN	1584	3216
	SAYFULLINA	3701	4
	GREEN	5136	3533
	SKILLSPAN (dev)	914	2260
	SAYFULLINA (dev)	1853	2
	GREEN (dev)	592	372
Test	Proprietary (expert)	1023	411
	Proprietary (cross-annot)	1534	937
	SKILLSPAN	974	2595
	SAYFULLINA	1849	2
	GREEN	233	102

C. Model Configuration

C.1. Binary Relevance Classification

The binary relevance classifier fine-tunes `roberta-base` with LoRA, later merged into a standalone checkpoint. Hyperparameters were selected with Optuna [29] (TPE sampler, median pruning, early stopping) to maximise validation macro-F1. The selected values are given in Table 8. Training uses cross-entropy loss and AdamW at fp16, with the best checkpoint restored by validation F1 and the decision threshold tuned on validation.

C.2. LLM Ranker Configuration

The Qwen3-14B-AWQ model is served locally through vLLM with the following configuration:

```
vllm serve Qwen/Qwen3-14B-AWQ \
  --quantization awq \
  --max-model-len 8192 \
  --port 8000 \
  --enable-prefix-caching \
  --max-num-seqs 100
```

Decoding is greedy (temperature 0) for deterministic output. Prefix caching is enabled to reuse the shared system prompt and demonstration prefix across candidate lists.

Table 8
LoRA fine-tuning configuration for the binary relevance classifier. Tuned hyperparameters were selected with Optuna over the listed search space.

Setting	Search space	Selected
Base model	n/a	`roberta-base`
Max sequence length	n/a	128
LoRA rank r	$\{4, 8, 16, 32\}$	4
LoRA α	$2r$	8
LoRA target modules	n/a	query, value
LoRA dropout	$[0.0, 0.3]$	0.205
LoRA bias	n/a	none
Learning rate	$[10^{-5}, 5\times10^{-4}]$	4.09×10^{-4}
Weight decay	$[0.0, 0.1]$	0.044
Warmup ratio	$[0.0, 0.1]$	0.012
Gradient accumulation	$\{1, 2, 4\}$	4
Epochs	n/a	8
Per-device batch size	n/a	8
Optimiser / precision	n/a	AdamW / fp16
Decision threshold	n/a	0.54

D. LLM Refinement of the Relevance Gate

As an optional precision lever, we route only the gate's uncertain predictions to an LLM auditor while trusting its confident ones. Sentences scoring below the decision threshold (0.54) are auto-rejected and those at or above an upper threshold ($t_{\text{upper}} = 0.90$) auto-accepted; only the uncertain band $[0.54, 0.90)$ is passed to the auditor, which re-decides each sentence's relevance. On the SKILL-XL test split, this band contains 381 of 2,641 sentences (14%), so the auditor is invoked sparingly. Table 9 reports the effect on the headline configuration.

Table 9
Effect of LLM gate refinement on the headline configuration (n0+m1, gated pipeline). The refinement trades end-to-end recall for deployment precision.

Metric	Plain gate	Refined gate
Gate F1 (SKILL-XL)	0.828	0.846
MAP Deploy@5	0.664	0.717
F1 (all)	0.249	0.266
MAP Standard@5	0.395	0.352

E. Full Configuration Results

Table 10 reports end-to-end performance across all nine prompt configurations at $k = 5$, for both the ungated and gated pipelines. Here n denotes the number of skill-bearing

Table 10
End-to-end performance across all prompt configurations and retrievers (plain gate, $k = 5$); n and m denote the number of skill-bearing and non-skill-bearing demonstrations respectively.

Retriever	Pipeline	Config	$F1_{all}$	$F1_{sb}$	avg_preds	$MAP_{Standard}$	MAP_{Deploy}
CurriculumMatch	ungated	Zero-shot	0.1505	0.2963	7.19	0.4076	0.2203
		n0+m1	0.1603	0.3002	6.91	0.4122	0.2814
		n0+m3	0.1618	0.2867	7.15	0.4174	0.3400
		n1+m0	0.1129	0.2307	13.07	0.4224	0.2793
		n1+m1	0.1359	0.2536	9.92	0.4195	0.3469
		n2+m3	0.1279	0.2471	11.02	0.4292	0.3158
		n3+m0	0.1101	0.2289	13.66	0.4317	0.2759
		n3+m2	0.1247	0.2386	11.74	0.4316	0.3391
		n3+m3	0.1205	0.2364	12.03	0.4312	0.3207
	gated	Zero-shot	0.2444	0.2939	3.56	0.3890	0.6574
		n0+m1	0.2493	0.2980	3.62	0.3949	0.6641
		n0+m3	0.2368	0.2849	4.12	0.4013	0.6690
		n1+m0	0.1847	0.2299	6.99	0.4053	0.6719
		n1+m1	0.2050	0.2526	5.74	0.4032	0.6704
		n2+m3	0.1993	0.2465	6.17	0.4124	0.6707
		n3+m0	0.1803	0.2278	7.34	0.4159	0.6678
		n3+m2	0.1914	0.2378	6.75	0.4146	0.6730
		n3+m3	0.1879	0.2360	6.80	0.4154	0.6710
ConTeXTMatch	ungated	Zero-shot	0.1410	0.2792	6.87	0.3904	0.2077
		n0+m1	0.1452	0.2711	6.81	0.3925	0.2850
		n0+m3	0.1324	0.2583	8.27	0.3964	0.2738
		n1+m0	0.1014	0.2131	12.90	0.4073	0.2675
		n1+m1	0.1216	0.2248	10.03	0.4015	0.3381
		n2+m3	0.1113	0.2203	11.52	0.4083	0.2970
		n3+m0	0.1013	0.2099	13.03	0.4133	0.2810
		n3+m2	0.1102	0.2139	11.85	0.4064	0.3251
		n3+m3	0.1063	0.2125	12.39	0.4057	0.3002
	gated	Zero-shot	0.2343	0.2790	3.28	0.3743	0.6529
		n0+m1	0.2264	0.2696	3.57	0.3780	0.6603
		n0+m3	0.2124	0.2582	4.31	0.3804	0.6551
		n1+m0	0.1699	0.2127	6.71	0.3916	0.6652
		n1+m1	0.1847	0.2243	5.76	0.3865	0.6645
		n2+m3	0.1762	0.2200	6.34	0.3916	0.6621
		n3+m0	0.1659	0.2095	6.98	0.3966	0.6598
		n3+m2	0.1705	0.2134	6.74	0.3906	0.6637
		n3+m3	0.1692	0.2122	6.83	0.3901	0.6616

demonstrations and m the number of non-skill-bearing (rejection) demonstrations. The GitHub repository describing the entire evaluation pipeline can be found here: https://github.com/AleksanderB-hub/Ranking_Pipeline_Final