أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

SkillRAG: Skill-Aware and Explainable CV Retrieval and Ranking

Abdulaziz Alqurashi, **Moaaz Noruuddin**, **Ahmed Radwan**

abdulaziz.alqurashi@kaust.edu.sa, moaaz.aminulhaque@kaust.edu.sa, ahmedyradwan02@gmail.com

Abstract

Screening large candidate pools against a job description is a core bottleneck in recruitment pipelines. Keyword matching and generic semantic similarity often fail to capture the specific skills a role requires, scale poorly to large collections, and offer little insight into *why* a candidate was selected. We present **SkillRAG**, a skill-aware and explainable CV retrieval system that extracts required skills from a job description, uses them to filter the candidate pool, condenses retained CVs into a standardized representation using a large language model, ranks candidates with a semantic embedding model, and generates evidence-based post-hoc explanations for the Top-10 shortlist. We evaluate on the TalentCLEF job-to-candidate retrieval benchmark. An encoder ablation across lexical and neural retrieval methods identifies Qwen3-Embedding-0.6B as the strongest bi-encoder for this task, reaching nDCG@10 of 0.978 while requiring only a single forward pass per document. Skill-based filtering combined with LLM condensation reduces the token footprint of candidate representations by 9.8× at a cost of 0.014 nDCG@10, making the efficiency gain close to quality-neutral. An independent judge model rates the resulting explanations 4.58 out of 5 on helpfulness. We outline future work on richer skill extraction, stronger faithfulness guarantees for explanations, and fairness-aware ranking.

Date: August 24, 2026

Code: https://github.com/AzizQri/SkillRAG

Contents

1 Introduction **2**

2 Related Work **3**

2.1 Pretrained Language Models for Document and Passage Ranking . . . 3

2.2 Recruitment and Job-Candidate Matching . . . 3

2.3 Fairness-Aware Recruitment Ranking . . . 3

2.4 Retrieval-Augmented Generation and LLM-Based Evaluation . . . 4

2.5 Research Gap and Positioning . . . 4

3 Methodology **4**

3.1 Pipeline Overview . . . 4

3.2 Dataset . . . 6

3.3 Implementation Details . . . 7

3.4 Evaluation Protocol . . . 7

4 Results **7**

4.1 Encoder Evaluation . . . 7

4.2 Effect of Skill Filtering and CV Condensation . . . 8

4.3 Explainability Results . . . 9

4.4 Summary of Findings . . . 9

5 Discussion **10**

6 Limitations and Future Work **11**

6.1 Limitations . . . 11

6.2 Future Work . . . 12

7 Conclusion **13**

References **13**

1 Introduction

Recruitment teams routinely screen large numbers of curricula vitae (CVs) against a single job description (JD) to produce a shortlist of promising candidates. As application volume grows, manual screening becomes time-consuming, inconsistent across reviewers, and difficult to audit, motivating a long line of automated CV retrieval and ranking approaches.

Existing automated approaches fall into three broad categories. Keyword and lexical matching methods compare a job description and a CV through overlapping terms [1]; they are computationally inexpensive but brittle to synonyms, paraphrasing, and implicitly described skills. Generic semantic similarity methods embed the job description and CV as dense vectors and compare them directly [2, 3]; these are more robust to lexical variation, but a single whole-document embedding obscures which specific requirements a candidate does or does not satisfy. End-to-end LLM processing, in which a language model reads the full JD and full CV for every comparison, captures nuanced requirement-level reasoning but is costly and difficult to scale to large candidate pools [4].

These approaches share a further limitation: a lack of transparency. Even when a ranking is produced, it is often unclear to a recruiter which parts of a CV drove a candidate's position, or whether the requirements most important to the role were considered at all. In a domain where ranking outputs materially affect candidates' opportunities, this opacity is a substantive problem rather than a cosmetic one.

Proposed system. We present SkillRAG, a skill-aware and explainable CV retrieval and ranking system. Rather than comparing a job description and a CV as two undifferentiated blocks of text, SkillRAG first extracts the required skills from the job description, uses them to filter the candidate pool, normalizes surviving CVs into a standardized condensed representation, performs embedding-based semantic retrieval over that representation, and produces an evidence-based explanation for each retrieved candidate.

Contributions. This report makes the following contributions:

- An encoder ablation across lexical and neural retrieval methods on the TalentCLEF benchmark, identifying Qwen3-Embedding-0.6B as the strongest of the models evaluated (nDCG@10 = 0.978) and showing that a lightweight bi-encoder can outperform late-interaction alternatives on this task.
- A quantified efficiency–quality trade-off for skill-based filtering combined with LLM CV condensation: a 9.8× reduction in candidate representation size at a cost of 0.014 nDCG@10.
- An explanation layer whose quality is measured rather than assumed, evaluated by an independent judge model along groundedness and helpfulness, with explanations rated 4.58/5 on helpfulness.

Assumptions and scope. SkillRAG assumes that job requirements can be reasonably approximated by a set of extractable skill keywords, that CVs contain sufficient information to be condensed into a standardized representation without losing what matters most to those requirements, and that a large judge model is a useful if imperfect proxy for human evaluation of explanation quality. The system is evaluated on a single benchmark (TalentCLEF) at a fixed retrieval cutoff (Top-10); claims about generalization to other domains, languages, or hiring contexts are outside the scope of these experiments.

The remainder of this report is organized as follows. Section 2 reviews related work and positions SkillRAG relative to it. Section 3 describes the pipeline and evaluation procedure. Section 4 reports results on TalentCLEF, and Section 5 discusses them. Section 6 covers limitations

and future work; Section 7 concludes.

2 Related Work

This section situates SkillRAG within four lines of work: pretrained-model-based document retrieval and ranking; recruitment and job-candidate matching; fairness in recruitment ranking; and retrieval-augmented generation together with LLM-based evaluation of generated text.

2.1 Pretrained Language Models for Document and Passage Ranking

The introduction of large pretrained transformer language models substantially changed the design of retrieval and ranking systems. BERT [5] demonstrated that a deep bidirectional transformer, pretrained on large unlabeled corpora and fine-tuned on a downstream task, could produce strong contextual representations of text. Building directly on this, Nogueira and Cho [6] showed that a BERT-based cross-encoder, applied to re-rank an initial set of retrieved passages, substantially improved passage ranking quality over classical lexical baselines such as BM25. Nogueira et al. [7] extended this idea to document ranking using a pretrained sequence-to-sequence model, framing relevance ranking as a generation task and showing further gains over purely discriminative rankers.

A complementary line of work encodes queries and documents independently into a shared dense vector space, allowing document representations to be precomputed and retrieved by approximate nearest-neighbor search [2, 3]. Such bi-encoders forgo the query–document interaction available to cross-encoders, but scale to large collections at a fraction of the inference cost. This is the architecture SkillRAG ultimately adopts, following the encoder ablation in Section 4.1.

These works establish two ideas that SkillRAG builds on: (i) that pretrained transformer representations are substantially more effective than lexical overlap for capturing semantic relevance, and (ii) that a two-stage design consisting of a lightweight retrieval stage followed by a more expensive re-ranking or scoring stage is an effective way to balance retrieval quality against computational cost. SkillRAG follows a similar two-stage structure, using a keyword-based filtering stage prior to a heavier embedding-based and RAG-based scoring stage, but differs in that its lightweight stage is explicitly skill-aware rather than a generic first-stage retriever such as BM25.

2.2 Recruitment and Job-Candidate Matching

A parallel line of work applies semantic matching to recruitment, where the compared documents are a job description and a candidate CV rather than a query and a passage. Contrastive learning approaches to resume–job matching [8] fine-tune encoders on paired job–CV data using data augmentation and contrastive objectives, showing that domain-specific training improves on generic sentence embeddings for this task. A recurring limitation in this literature is reliance on proprietary or hand-labeled relevance judgments, which constrains reproducibility and cross-study comparison.

SkillRAG takes the opposite trade-off. It does not fine-tune on proprietary labeled data; instead it evaluates publicly available pretrained encoders on the public TalentCLEF benchmark [9] and selects among them by measured retrieval quality (Section 4.1). This favors reproducibility at the likely cost of task-specific gains a fine-tuned model would achieve.

2.3 Fairness-Aware Recruitment Ranking

Automated recruitment systems can encode or amplify demographic biases present in training data or in the documents themselves [10]. Benchmarking frameworks [11] have been proposed to

systematically measure gender and other demographic bias in automated hiring and ranking, motivating treatment of fairness as an explicit evaluation axis rather than an incidental property of a ranking system.

SkillRAG does not currently implement fairness-aware ranking or reranking; the pipeline scores candidates on skill coverage and semantic relevance only. We note this literature here because it bears directly on how a system of this kind should be evaluated before deployment, and because organizing a ranker around explicit skill requirements does not by itself guarantee fair outcomes. Fairness-aware reranking is discussed as future work in Section 6.

2.4 Retrieval-Augmented Generation and LLM-Based Evaluation

Retrieval-Augmented Generation (RAG) [12] combines a non-parametric retrieval component with a parametric generation component, conditioning generated text on retrieved evidence rather than relying solely on parameters learned during pretraining. This general pattern—retrieve relevant evidence, then condition generation on that evidence—underlies both the RAG-based scoring stage and the post-hoc explanation stage of SkillRAG: the scoring and explanation LLM calls are conditioned on the job description together with the specific retrieved and condensed CV evidence for a candidate, rather than being asked to reason about relevance from an unconditioned prompt.

Because RAG-generated text is only as reliable as its support in the retrieved evidence, groundedness has become a standard evaluation axis for RAG systems [13]. Evaluating such properties at scale has increasingly relied on strong language models acting as judges [14]. This approach correlates reasonably with human preference judgments but carries documented biases, including preferences for longer responses, position effects, and self-preference when the judge shares a family with the generator [14]. SkillRAG adopts groundedness and helpfulness as its two explanation-quality metrics (Section 3.4), evaluated by a judge model from a different family than the explanation generator, which mitigates but does not eliminate these concerns.

2.5 Research Gap and Positioning

Existing work largely addresses these problems in isolation: neural ranking methods target semantic relevance; recruitment-matching methods target domain adaptation of embeddings; fairness benchmarks target bias measurement; and RAG methods target evidence-conditioned generation, typically in open-domain question answering rather than recruitment. What is less common is the combination of skill-level requirement awareness with an explanation layer whose quality is itself measured, inside a single recruitment pipeline. SkillRAG's contribution is this integration, together with an explicit encoder ablation establishing which retrieval model the pipeline should be built on.

3 Methodology

3.1 Pipeline Overview

Figure 1 summarizes the SkillRAG pipeline, which consists of the following stages.

1. **Job description input.** The system receives a job description (JD) containing the requirements and skills relevant to candidate selection.
2. **Required skill extraction.** The job description is processed to identify required skills as keywords:

$$\text{Job Description} \longrightarrow \text{Required Skills as Keywords (Regex)}.$$

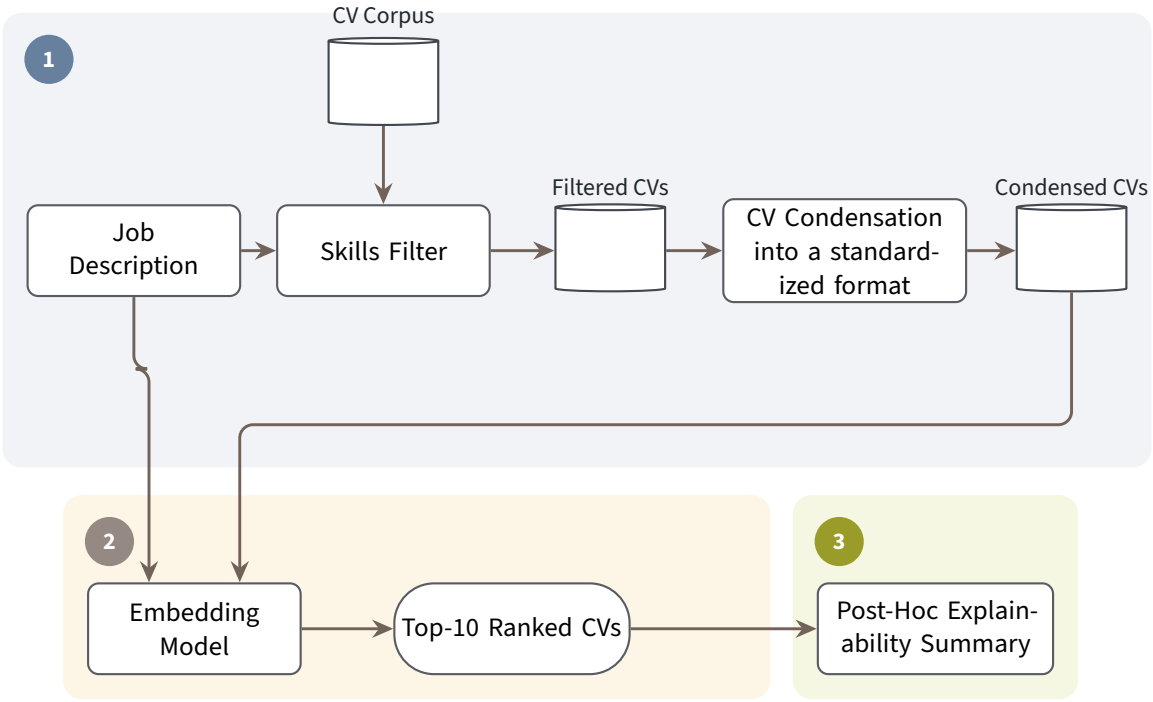


Figure 1 Overview of the SkillRAG pipeline. **(1)** Skills extracted from the job description filter the CV corpus, and the retained CVs are condensed into a standardized format. **(2)** The job description and the condensed CVs are embedded and compared, producing a Top-10 ranking. **(3)** Each shortlisted candidate receives a post-hoc explainability summary.

The extraction operates over a curated skill vocabulary of [N] terms, matched case-insensitively with [word-boundary matching / stemming / alias handling]. This stage produces a skill-focused representation of the job's requirements and enables the subsequent filtering step. It is a lightweight, rule-based mechanism rather than a learned skill extractor; its limitations are discussed in Section 6.

3. **CV filtering.** The extracted required skills are used together with the full CV collection to reduce the candidate pool:

$$\text{CV Collection} + \text{Required Skills} \longrightarrow \text{Filtered CVs}.$$

Formally, let S_{JD} denote the set of skills extracted from the job description and let S_c denote the set of skill terms detected in candidate c's CV. A candidate is retained if

$$\left|S_{\text{JD}} \cap S_c\right| \geq \tau. \tag{1}$$

We set τ = [VALUE] in all reported experiments, chosen [as the smallest value retaining $\geq$ [Y]% of relevant candidates on the development set]. This stage reduces the mean candidate pool from [A] to [B] CVs per job description. It is an initial filtering step intended to reduce the search space, not the final ranking mechanism.

4. **CV condensation / normalization.** Filtered CVs are passed to Gemini 2.5 Flash, which maps each CV into a consistent, one-page representation following a standardized structure:

$$\text{Filtered CV} \xrightarrow{\text{Gemini 2.5 Flash}} \text{Condensed CV}.$$

CVs submitted by real candidates vary widely in structure, length, formatting, and level of detail, which makes direct semantic comparison difficult. Condensing CVs into a

standardized representation is intended to reduce irrelevant formatting variation, preserve the professional information most relevant to matching, and improve the consistency of downstream embedding-based retrieval. This normalization step is not assumed to be lossless; possible information loss is discussed as a limitation in Section 6.

5. **Embedding-based retrieval.** The job description and each condensed CV are embedded using a semantic embedding model, and candidates are ranked by embedding-space similarity:

$$\text{Job Description + Condensed CV} \xrightarrow{\text{Embedding Model}} \text{Candidate Ranking.}$$

The underlying comparison is a bi-encoder cosine similarity between a query representation q (job description) and a candidate representation c:

$$\text{sim}(q, c) = \frac{q \cdot c}{\|q\| \, \|c\|}. \tag{2}$$

The embedding model used in the downstream pipeline, Qwen3-Embedding-0.6B [15], was selected through the encoder ablation reported in Section 4.1.

6. **Top-10 retrieval and RAG-based scoring.** Retrieved candidates are re-scored by conditioning [MODEL] on the job description together with the candidate's condensed CV:

$$\text{Job Description + Condensed CV} \longrightarrow \text{RAG Score} \longrightarrow \text{Top-10.}$$

The model returns [a scalar relevance score in range R / a structured per-requirement judgment], combined with the embedding similarity as [FORMULA]. Top-10 is the retrieval cutoff evaluated in these experiments; it is not claimed to be optimal for all deployment settings.

7. **Post-hoc explanation.** For each Top-10 candidate, an explanation is generated using Gemini 2.5 Flash, conditioned on the candidate's condensed CV and RAG score:

$$\text{Top-10 Candidate + Condensed CV + RAG Score} \xrightarrow{\text{Gemini 2.5 Flash}} \text{Explanation.}$$

Explanations focus on evidence-based reasons for candidate relevance: the relevant skills identified, supporting evidence drawn from the candidate's CV, the relationship between this evidence and the job requirements, candidate strengths, and, where supported, missing or weaker requirements. Because a fluent or plausible-sounding explanation is not necessarily faithful to the underlying evidence, explanation quality is measured rather than assumed, as described in Section 3.4.

3.2 Dataset

We evaluate on TalentCLEF [9], a benchmark for job-to-candidate retrieval. The [split name] split contains [N_JD] job descriptions and [N_CV] candidate profiles in [language(s)], with a mean of [M] relevant candidates per job description (range [min]–[max]).

The ratio between relevant-set size and retrieval cutoff bears directly on how the reported metrics should be read. With a mean of [M] relevant candidates per query and a cutoff of 10, Recall@10 is bounded above by approximately $10/[M] \approx$ [CEILING]. Reported Recall@10 values should therefore be interpreted relative to this ceiling rather than against 1.0. nDCG@10, which measures the ranking quality of the retrieved set, is not subject to this constraint and is the primary metric throughout.

3.3 Implementation Details

Embeddings are computed with Qwen3-Embedding-0.6B ([D]-dimensional, maximum sequence length [L]; inputs exceeding this are [truncated / chunked]). Retrieval uses [exact cosine similarity / FAISS index type]. CV condensation and explanation generation use Gemini 2.5 Flash at temperature [T]; explanation evaluation uses GPT-5-mini at temperature [T]. Experiments ran on [hardware]; the full pipeline over the evaluation set required approximately [time / cost].

3.4 Evaluation Protocol

Evaluation is organized into two components.

A. Encoder evaluation. Candidate embedding models for the retrieval stage are compared on TalentCLEF using nDCG@10 and Recall@10, together with a qualitative measure of computational cost (forward passes per document, use of an approximate nearest-neighbor index, or a shared forward pass across multiple scoring heads). The best-performing encoder is selected for the downstream pipeline (Section 4.1).

B. Explainability evaluation. Post-hoc explanations produced by Gemini 2.5 Flash for the retrieved Top-10 candidates are evaluated by an independent judge model, GPT-5-mini, on two axes:

- **Groundedness [0–1]:** the extent to which claims in the explanation are supported by the candidate's condensed CV, as opposed to containing unsupported assertions.
- **Helpfulness [1–5]:** the usefulness, relevance, and overall quality of the explanation for understanding why a candidate was retrieved and ranked.

[N] explanations were scored ([N_JD] job descriptions × 10 candidates); reported figures are means across all scored explanations. The judge receives [the job description, the condensed CV, and the explanation] and returns [scores only / scores with written justification].

Using a judge from a different model family than the explanation generator avoids the self-preference bias documented in Section 2.4, though it does not eliminate other known judge biases. This evaluation is kept distinct from the encoder evaluation, since it targets the quality of generated text rather than the quality of a ranking.

Baseline versus skill-aware pipeline. In addition to the encoder ablation, we compare a baseline configuration, in which the selected encoder is applied directly to full, unfiltered, uncondensed CVs, against the SkillRAG pipeline, which applies required-skill filtering, CV condensation, and the same encoder. This comparison isolates the effect of skill-aware filtering and standardized CV representation on retrieval quality, and quantifies the accompanying change in representation cost via token reduction.

4 Results

Results for Qwen3-Embedding-4B are omitted; those runs did not complete within the project timeline. All figures come from a single run per configuration; no variance estimates or significance tests are reported, and small differences between configurations should be read accordingly.

4.1 Encoder Evaluation

Table 1 reports lexical and learned-sparse first-stage retrieval methods; Table 2 reports neural encoders.

Among lexical and sparse methods, SPLADE-v3 [16] outperforms both BM25 [1] variants on nDCG@10 and Recall@10, at the cost of one forward pass per document. Reducing the job

Table 1 Lexical and learned-sparse first-stage retrieval baselines.

Model	Type	nDCG@10	Recall@10	Inference cost
BM25 (bm25s), full JD as query	lexical	0.8941	0.2763	none (index-time only)
BM25, JD reduced to requirement terms	lexical	0.8891	0.2763	none (index-time only)
SPLADE-v3 (naver/splade-v3)	learned sparse	0.9278	0.2847	1 fwd/doc

description to requirement terms slightly decreased nDCG@10 for BM25 (0.8941 to 0.8891) with identical Recall@10, indicating that lexical matching over the full JD text was not harmed by the additional non-requirement content.

Table 2 Neural encoder evaluation. Qwen3-Embedding-4B is omitted (incomplete runs).

Model	Interaction	nDCG@10	Recall@10	Inference cost
Qwen3-Embedding-0.6B	bi-encoder	0.9779	0.2982	1 fwd/doc + ANN
jina-colbert-v2	late	0.9554	0.2921	1 fwd/doc + late interaction
BGE-M3 (sparse head)	lexical	0.9485	0.2919	shared fwd/doc
BGE-M3 (dense head)	bi-encoder	0.9431	0.2848	shared fwd/doc
BGE-M3 (ColBERT head)	late	0.9145	0.2759	shared fwd/doc

Among neural encoders, Qwen3-Embedding-0.6B achieves the highest nDCG@10 (0.9779) and Recall@10 (0.2982), ahead of jina-colbert-v2 [17] and all three BGE-M3 [18] scoring heads, despite being a comparatively lightweight bi-encoder requiring only one forward pass per document plus approximate nearest-neighbor search. That a bi-encoder outperforms a late-interaction model on this benchmark is notable, since late interaction is generally expected to improve ranking precision at higher inference cost. On this basis, Qwen3-Embedding-0.6B was selected as the embedding model for the downstream SkillRAG pipeline.

The two metrics behave very differently across methods. nDCG@10 spans 0.8891 to 0.9779, whereas Recall@10 varies only between 0.2763 and 0.2982 across lexical, learned-sparse, dense, and late-interaction architectures alike. This is consistent with the recall ceiling described in Section 3.2: with a mean of [M] relevant candidates per job description, no method can exceed approximately [CEILING] at a cutoff of 10. The methods evaluated therefore differ mainly in how well they order the retrieved candidates, not in how many relevant candidates reach the top 10. nDCG@10 is accordingly the discriminating metric throughout this section.

4.2 Effect of Skill Filtering and CV Condensation

Table 3 compares each encoder's performance on the unfiltered, uncondensed baseline representation against the filtered and condensed representation produced by the SkillRAG pipeline.

For Qwen3-Embedding-0.6B, skill-based filtering and CV condensation reduce the mean candidate representation from [A] to [B] tokens (a factor of 9.8×), at a cost of 0.014 nDCG@10 (0.9779 to 0.9640) and 0.005 Recall@10 (0.2982 to 0.2930). For jina-colbert-v2 the token reduction is smaller (4.1×) and retrieval quality moves in the opposite direction: nDCG@10 rises by 0.017 (0.9554 to 0.9720) and Recall@10 by 0.003 (0.2921 to 0.2950).

The two encoders therefore respond to the same condensed representation in opposite directions, and the gain for the late-interaction model is slightly larger in magnitude than the loss for the bi-encoder. One reading is that normalizing CVs into a consistent structure benefits

Table 3 Effect of skill-aware filtering and LLM-based CV condensation on retrieval quality and representation size. Only the two strongest encoders from Table 2 were evaluated under this comparison; Qwen3-Embedding-4B is omitted (incomplete runs).

Model	nDCG@10	Recall@10	Mean tokens/CV	Reduction
Qwen3-Embedding-0.6B (baseline)	0.9779	0.2982	[A]	—
Qwen3-Embedding-0.6B (filtered, condensed)	0.9640	0.2930	[B]	9.8×
jina-colbert-v2 (baseline)	0.9554	0.2921	[C]	—
jina-colbert-v2 (filtered, condensed)	0.9720	0.2950	[D]	4.1×

token-level matching specifically: late interaction scores query and document terms against each other directly, so it stands to gain from a representation in which comparable information appears in comparable places. We did not test this, and doing so would require running the full encoder set through both representations.

A consequence worth stating plainly is that on the condensed representation—the representation the deployed pipeline actually uses—jina-colbert-v2 (0.9720) outperforms Qwen3-Embedding-0.6B (0.9640), reversing the ordering observed on the baseline representation in Section 4.1. The encoder selection reported there was made on unfiltered, uncondensed CVs, and it does not survive the change of representation.

The difference in achievable compression between the two configurations is a property of the condensation stage rather than of the encoders, since both consume the same condensed CVs. It reflects [differing tokenizers / differing maximum sequence lengths, which bound how much of each original CV was encoded in the baseline condition].

4.3 Explainability Results

Table 4 reports the GPT-5-mini judge evaluation of the post-hoc explanations generated by Gemini 2.5 Flash for the SkillRAG Top-10 shortlist.

Table 4 Evaluation of SkillRAG post-hoc explanations, judged by GPT-5-mini over [N] explanations.

Metric	Score
Groundedness [0–1]	0.97
Helpfulness [1–5]	4.58

Helpfulness is rated 4.58 out of 5, indicating that the judge model found the generated explanations generally useful and relevant for understanding candidate relevance. Groundedness scored 0.97, which would indicate that explanations were on the whole well supported by the retrieved candidate evidence.

That figure, however, rests on a single unvalidated measurement. Groundedness was scored by one judge model against a rubric that was never calibrated against human annotation, and a near-ceiling result is also what an insufficiently discriminating rubric would produce; nothing in our setup separates those two cases. We therefore treat the groundedness figure as provisional, and Section 5 interprets the two metrics accordingly.

4.4 Summary of Findings

Qwen3-Embedding-0.6B provided the strongest retrieval quality of the encoders evaluated on the baseline representation, while remaining computationally lightweight relative to interaction-

heavier alternatives. Skill-aware filtering combined with LLM-based condensation substantially reduced representation size, at a small cost in ranking quality for the bi-encoder and a small gain for the late-interaction model, which overtakes it on the condensed representation. Explanations were rated highly on helpfulness, while the groundedness figure carries the caveat noted above.

5 Discussion

Encoder choice. The ablation in Section 4.1 suggests that skill-aware retrieval benefits from a semantically rich but comparatively lightweight bi-encoder. Qwen3-Embedding-0.6B outperformed both a late-interaction model (jina-colbert-v2) and all three BGE-M3 scoring heads while requiring only a single forward pass per document plus approximate nearest-neighbor search. This runs against the usual expectation that late interaction buys ranking precision at the cost of inference, and is consistent with the broader observation in the retrieval literature (Section 2.1) that well-trained dense bi-encoders can be highly competitive as primary retrievers. That advantage is specific to the baseline representation, however. As Section 4.2 shows, jina-colbert-v2 improves once CVs are normalized into a consistent structure and overtakes Qwen3-Embedding-0.6B on that representation, so the gap measured in Section 4.1 is better read as a difference in how well each model copes with heterogeneous raw CVs than as a general advantage of bi-encoders over late interaction. Establishing which reading is right would require evaluating the full encoder set across both representations, which we did not do.

What the metrics can and cannot distinguish. The near-constancy of Recall@10 across architectures (0.2763–0.2982) alongside a much wider nDCG@10 spread (0.8891–0.9779) indicates that these methods differ in how they order candidates rather than in which candidates they surface. Given the recall ceiling of approximately [CEILING] at this cutoff (Section 3.2), all evaluated methods retrieve a similar fraction of the relevant set; the differences lie entirely in ranking quality within it.

This also bears on how the absolute numbers should be read. An nDCG@10 of 0.978 is high in absolute terms, and we do not interpret it as evidence that job-to-candidate matching is close to solved. TalentCLEF's relevance judgments are constructed around [occupational categories / job-title alignment], which may make the discrimination required easier than the fine-grained distinctions a recruiter draws between candidates who are all nominally qualified. Performance on this benchmark should be read as evidence that the pipeline functions as designed, not as a measure of real-world screening quality.

Efficiency and its accounting. The effect of skill-based filtering and CV condensation (Section 4.2) is best read as an efficiency–quality trade-off rather than a straightforward improvement. Retrieval quality after condensation falls slightly for Qwen3-Embedding-0.6B and rises for jina-colbert-v2, so for the bi-encoder the case for condensation rests on the 9.8× reduction in representation size rather than on a ranking gain; for the late-interaction model it does not have to, since condensation improves its ranking as well as shrinking its input.

That reduction is not free. Condensation requires one LLM call per CV before any retrieval occurs, so for a single job posting against a fresh candidate pool the pipeline costs more than embedding raw CVs directly. The economics invert when the same condensed pool is queried repeatedly: condensation is a one-time amortized cost, whereas the token savings recur at every query. Skill-based filtering compounds this effect, since only candidates passing the threshold are condensed at all. For an organization screening many job postings against a stable candidate database, this is the regime in which the design pays off; for one-off screening, it is not.

Relative importance of design choices. The spread in nDCG@10 across the neural encoders evaluated (0.063) is larger than the change induced by filtering and condensation for a fixed encoder (0.014 for Qwen3-Embedding-0.6B, 0.017 for jina-colbert-v2). On this benchmark, encoder selection therefore appears to matter more for retrieval quality than the decision to filter and condense, which supports treating the encoder ablation as a prerequisite to evaluating pipeline-level design choices. This comparison rests on two models under the filtering condition, however, and a firmer claim would require running the full encoder set through both representations.

Explainability. The helpfulness score (4.58/5) indicates that the judge model found the generated explanations useful, which is a reasonable proxy for recruiter utility but remains a proxy: no human evaluation was conducted, and helpfulness as judged by a language model may reward fluency and structure over decision-relevance. The groundedness result, subject to the caveat noted in Section 4.3, would if reliable indicate that explanations are substantively tied to the retrieved evidence.

The distinction between the two axes is the point of the two-metric design. An explanation can be highly helpful—well-organized, specific, confident—while asserting things the CV does not support, and a recruiter reading it has no way to tell. This is why helpfulness alone would be an inadequate evaluation, and why the groundedness measurement needs to be reliable rather than approximately right. In a recruitment setting, where ranking and explanation outputs materially affect candidates' opportunities, an ungrounded explanation is not merely a quality defect but a mechanism for laundering an unjustified ranking into something that appears justified.

Information loss under condensation. Condensation carries a risk that the aggregate metrics do not capture: mapping a CV into a fixed, standardized structure may drop details relevant to a specific job's less common requirements. Aggregate nDCG can remain high while individual candidates are systematically disadvantaged by what the condensation stage chose to omit—for instance, if unusual qualifications are compressed away more readily than conventional ones. Detecting this would require per-candidate analysis of what condensation removes, which we did not perform.

6 Limitations and Future Work

6.1 Limitations

Evidence strength. All reported figures come from a single run per configuration, without variance estimates or significance testing. The differences discussed between filtering configurations (0.014 nDCG@10) are small enough that run-to-run variation cannot be ruled out as an explanation. Comparisons between encoders involve larger gaps and are correspondingly more robust, but the same caveat applies.

Single benchmark and cutoff. The system is evaluated on one benchmark (TalentCLEF), in [language], at one retrieval cutoff (Top-10). We do not know how the pipeline behaves on other domains, other languages, other candidate pool sizes, or at cutoffs a recruiter might actually use. As noted in Section 5, the benchmark's relevance judgments may also make the task easier than real-world screening.

Regex-based skill extraction. The extraction stage relies on keyword and regex matching over a fixed vocabulary. It may miss implicit skills, synonyms, unusual phrasings, or requirements implied by a combination of stated responsibilities rather than named explicitly. Because extraction

feeds the filtering stage, a missed skill can remove a qualified candidate from the pool entirely, before any semantic matching occurs; extraction errors are unrecoverable downstream.

LLM-based CV condensation. Using Gemini 2.5 Flash to normalize CVs into a standardized representation may introduce information loss, summarization errors, inconsistent transformations across candidates, or hallucinated details absent from the original. The aggregate retrieval metrics in Section 4.2 cannot detect any of these at the level of individual candidates; a CV could be badly condensed while overall nDCG remains high.

Embedding dependence. Retrieval quality depends directly on the selected embedding model and on how well it represents the semantics of both job requirements and candidate evidence. The ablation in Section 4.1 establishes that Qwen3-Embedding-0.6B was the strongest of the models tested on this benchmark, not that it is optimal in general.

Explanation evaluation. Both explanation metrics come from a language model, with no human validation. We do not know whether the judge's notion of helpfulness corresponds to what a recruiter would find useful, and the groundedness result carries the additional caveat described in Section 4.3. Post-hoc explanations also describe a ranking decision rather than causing it: the explanation model receives the candidate and its score, so a fluent explanation can be produced for a ranking that was itself mistaken.

Fairness. The system does not implement or evaluate fairness-aware ranking, and no demographic analysis of its outputs was performed. Two components warrant particular scrutiny in future work. Regex-based extraction over a fixed vocabulary may favor candidates who describe their experience in conventional terminology, disadvantaging equivalent experience expressed differently—a plausible correlate of educational or national background. LLM-based condensation makes an unaudited per-candidate decision about which details survive into the representation used for ranking. Organizing a system around explicit skill requirements does not make it fair, and may relocate bias into stages that are harder to inspect than a similarity score.

Operational cost. We report token reduction as a proxy for efficiency but do not measure wall-clock latency, monetary cost per query, or throughput on pool sizes larger than those evaluated. The efficiency argument in Section 5 is therefore directional rather than quantified.

6.2 Future Work

The most consequential gaps concern evaluation rather than architecture.

Strengthening the evidence. Repeated runs with variance estimates and significance testing; evaluation on additional datasets and at multiple retrieval cutoffs; and measurement of end-to-end latency and cost on larger candidate pools than those evaluated here.

Explanation faithfulness. Human evaluation of explanation quality against recruiter judgment; verification methods stronger than a single judge score, such as requiring explanations to cite specific CV spans that can be checked automatically against the source document; and resolution of the groundedness measurement issue noted in Section 4.3.

Fairness. Demographic bias measurement across the full pipeline, with attention to the extraction and condensation stages specifically; and fairness-aware reranking as a downstream component.

Pipeline improvements. Learned skill extraction to replace regex matching; requirement-aware retrieval beyond keyword-level filtering; evidence extraction linking specific CV spans to specific

requirements; and reduced dependence on per-candidate LLM calls, which currently dominate the pipeline's operating cost.

7 Conclusion

We presented SkillRAG, a skill-aware and explainable CV retrieval and ranking system that extracts required skills from a job description, filters and condenses candidate CVs, ranks them by semantic similarity, and generates evidence-based explanations for the resulting shortlist.

Three findings emerged from evaluation on TalentCLEF. First, a lightweight bi-encoder (Qwen3-Embedding-0.6B) outperformed both late-interaction and multi-head alternatives at a fraction of their inference cost, indicating that architectural sophistication in the retrieval stage was not what limited performance on this task. Second, skill-based filtering combined with LLM-based condensation reduced candidate representation size by 9.8× at a cost of 0.014 nDCG@10—a trade-off that favors deployments querying a stable candidate pool repeatedly, where condensation is amortized while token savings recur. Third, the two-axis explanation evaluation separated two properties that are easily conflated: explanations rated highly on helpfulness (4.58/5), while groundedness—whether those explanations are actually supported by the underlying CV—proved the harder property to measure reliably.

That third finding is the one we would emphasize. An explanation that reads well is not the same as an explanation that is true to the evidence, and in recruitment the difference matters: a well-written justification for a poorly ranked candidate makes a bad decision harder to challenge, not easier. Building explanation layers is straightforward; establishing that they are faithful is not, and the latter is where we believe work on explainable recruitment systems should concentrate.

References

[1] Stephen Robertson and Hugo Zaragoza. "The Probabilistic Relevance Framework: BM25 and Beyond". In: *Foundations and Trends in Information Retrieval* 3.4 (2009), pp. 333–389.

[2] Vladimir Karpukhin et al. "Dense Passage Retrieval for Open-Domain Question Answering". In: *Proceedings of EMNLP*. 2020, pp. 6769–6781.

[3] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of EMNLP-IJCNLP*. 2019, pp. 3982–3992.

[4] Weiwei Sun et al. "Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. arXiv:2304.09542. 2023.

[5] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of NAACL-HLT*. 2019, pp. 4171–4186.

[6] Rodrigo Nogueira and Kyunghyun Cho. "Passage Re-ranking with BERT". In: *arXiv preprint arXiv:1901.04085* (2019).

[7] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. "Document Ranking with a Pretrained Sequence-to-Sequence Model". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020, pp. 708–718.

[8] Xiao Yu, Jinzhong Zhang, and Zhou Yu. "ConFit: Improving Resume-Job Matching using Data Augmentation and Contrastive Learning". In: *Proceedings of the 18th ACM Conference on Recommender Systems (RecSys)*. 2024.

[9] Luis Gascó, Hermenegildo Fabregat, et al. “Overview of TalentCLEF: Skill and Job Title Intelligence for Human Capital Management”. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction (CLEF)*. 2026.

[10] Le Chen et al. “Investigating the Impact of Gender on Rank in Resume Search Engines”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI)*. 2018.

[11] Ze Wang et al. “JobFair: A Framework for Benchmarking Gender Hiring Bias in Large Language Models”. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. 2024.

[12] Patrick Lewis et al. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[13] Shahul Es et al. “RAGAS: Automated Evaluation of Retrieval Augmented Generation”. In: *Proceedings of EACL: System Demonstrations*. 2024.

[14] Lianmin Zheng et al. “Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2023.

[15] Yanzhao Zhang et al. “Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models”. In: *arXiv preprint arXiv:2506.05176* (2025).

[16] Carlos Lassance et al. “SPLADE-v3: New Baselines for SPLADE”. In: *arXiv preprint arXiv:2403.06789* (2024).

[17] Rohan Jha et al. “Jina-ColBERT-v2: A General-Purpose Multilingual Late Interaction Retriever”. In: *arXiv preprint arXiv:2408.16672* (2024).

[18] Jianlv Chen et al. “BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation”. In: *arXiv preprint arXiv:2402.03216* (2024).