Retrieve, Rerank, Survive: LLM-Listwise Pipelines for Job–Person and Job–Skill Matching at TalentCLEF 2026

bipboopbipboop at CLEF 2026 TalentCLEF

Leo Hemamou[1,*], Robin Dupont[1]

[1]*Independent Researcher*

Abstract

We describe team **bipboopbipboop**'s submissions to both shared tasks of TalentCLEF 2026. Task A (Contextualized Job–Person Matching) ranks candidate résumés by their relevance to a job offer, and Task B (Job–Skill Matching with Skill Type Classification) ranks ESCO skills relevant to a job title. We approach both tasks with a single unifying recipe: a *retrieval → pointwise reranking → listwise reranking* pipeline that couples a strong neural first stage, a pointwise neural reranker, and a large language model used as a listwise judge. The final listwise stage is tailored to each task. Our recurring empirical finding is that the gains of the LLM final stage are bounded by the *composition and depth of its input candidate set* rather than by its scoring ability: on Task B, fusing JobBERT-v3 candidates *before* the listwise stage improves validation graded NDCG from 0.7268 to 0.7734 (+6.4% relative), one of the most cost-effective improvements in the pipeline. On the official test sets, Task A reached an overall multilingual AVG_MAP of 0.6532 and a cross-lingual MAP of 0.6438; Task B reached a graded NDCG of 0.7793, where the system generalized slightly better on the held-out test set than on validation. We additionally report a consistent set of negative results—stacking listwise on listwise, skill-level rank fusion, retriever distillation, and AgentIR-style training—that did not improve over the pipeline.

Keywords

information retrieval, large language models, listwise reranking, LLM-as-a-judge, job-person matching, job-skill matching, ESCO, cross-lingual retrieval, pointwise reranking, TalentCLEF

1. Introduction

Matching people, jobs, and skills is a central problem in human-resources natural language processing (HR-NLP). The TalentCLEF lab series at CLEF provides shared benchmarks for Skill and Job Title Intelligence over the multilingual ESCO occupation and skill taxonomy [1, 2]. Its 2026 edition poses two information-retrieval tasks that share a common structure: a short, ambiguous query (a job offer or a job title) is ranked against a large candidate set (résumés or ESCO skills), and evaluation rewards the quality of the full ranked list rather than a single top hit.

Task A [3] ranks candidate résumés by relevance to a job offer (binary relevance, primary metric MAP) in three directions: monolingual English (en–en), monolingual Spanish (es–es), and cross-lingual (en–es). Task B [4] ranks the ESCO skills relevant to a job title, distinguishing *core* skills (graded relevance 2) from *contextual* skills (graded relevance 1); the primary metric is graded NDCG over a relevant set whose validation median is roughly 181 skills per query, so recall is as load-bearing as precision.

We treat both tasks with one methodology: a *retrieval → pointwise reranking → listwise reranking* pipeline sharing a strong neural first stage, a Qwen3-Reranker-8B pointwise reranker, and a Qwen3-235B-A22B-Instruct listwise judge, both served via the DeepInfra API. As a GPU-constrained team we deliberately push all heavy inference onto hosted APIs and rent a single GPU only briefly for retriever fine-tuning and encoding; this constraint, more than scale, shapes the design. The tasks differ only in the final stage: for Task A a tournament reranker seeded from a geometric-mean fusion over the top-60; for Task B a select-then-rank listwise reranker over the top-300. Our recurring lesson is that

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany

*Corresponding author.

l.hemamou@gmail.com (L. Hemamou); robin@dupont.cc (R. Dupont)

0000-0001-8763-9453 (L. Hemamou); 0009-0003-4541-5146 (R. Dupont)

the LLM final stage's gains are not bounded by its scoring ability but by the *composition and depth of its input candidate set.* On Task B, fusing a complementary title-semantic embedder (JobBERT-v3) into the candidate pool *before* the listwise stage raises validation graded NDCG from 0.7268 to 0.7734 (+6.4% relative): invest in recall and pool composition before listwise sophistication. On the official test sets, Task A reached AVG_MAP 0.6532 and Task B graded NDCG 0.7793. We tag every metric as development/validation or test and never interchange the two splits.

2. Related Work

HR-NLP tasks are increasingly grounded in the multilingual ESCO taxonomy [5]. The TalentCLEF lab consolidates several into shared retrieval benchmarks: the 2025 edition [1] and the 2026 edition extending them to job–person matching (Task A, [3]) and job–skill matching (Task B, [4]), described in the 2026 overview [2]. Multi-stage retrieval—a fast first-stage retriever feeding a more expensive reranker—is standard practice. Our first stages use the Qwen3-Embedding dense encoders [6] adapted with LoRA [7], our mid stage the Qwen3-Reranker-8B reranker [8], and for Task B JobBERT-v3 [9], a language-agnostic encoder specialized for job-title semantics, as a complementary signal.

LLMs are strong listwise rerankers. RankGPT [10] established the sliding-window paradigm our Task B select-then-rank stage follows; we also evaluate TourRank [11], which aggregates rankings across shuffled tournament rounds without elimination. A complementary line uses LLMs as relevance judges (*LLM-as-a-judge*), the view our Task A rubric stage adopts. For reranker fine-tuning we use ms-swift [12] and explored AgentIR-style [13] reasoning-augmented training.

3. Task A: Contextualized Job–Person Matching

Given a free-text job offer, Task A ranks a pool of candidate résumés by relevance [3, 2]. Our submission scores a candidate set in parallel by complementary matchers, combines their per-job-normalised scores by geometric-mean fusion, and reorders the fused top candidates with a listwise tournament.

3.1. Data

The task is evaluated in three directions: monolingual English (EN-EN), monolingual Spanish (ES-ES), and cross-lingual (EN-ES, an English job offer against a Spanish résumé corpus). The development set provides 10 queries per direction against a pool of 472 résumés each, with binary relevance judgements. The official test set is larger and harder (∼40 EN-EN queries dominated by Senior/Principal and domain-fused titles); EN-ES has no development qrels. The primary metric is MAP; secondary metrics are MRR, NDCG, and precision at cut-offs (P@5, P@10, P@100). We report development (DEV) and test (TEST) numbers separately.

3.2. System

The submitted system (Figure 1) seeds the fused top-60 into a listwise tournament reranker. The EN-EN direction fuses four matchers; the ES-ES and EN-ES directions fuse two.

Pointwise reranking. Every candidate is scored independently by Qwen3-Reranker-8B [8] served via DeepInfra (standalone development EN-EN MAP 0.6969).

Rubric LLM-as-a-judge. For each EN-EN query we auto-generate a job-specific rubric of roughly 220 yes/partial/no questions (scored 1.0/0.5/0.0) in 12 weighted categories. Each résumé is scored against this rubric by Qwen3-235B-A22B-Instruct-2507 (standalone EN-EN MAP 0.7791). The rubric is used only in the EN-EN chain; adding it to the Spanish chains hurts fusion. A sample rubric question is shown in Appendix A.1.

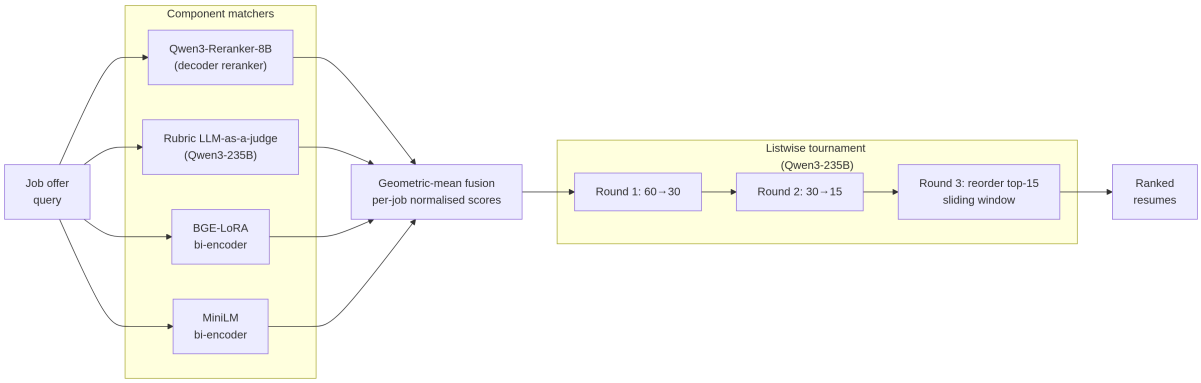

Figure 1: Submission pipeline for TalentCLEF 2026 Task A (EN-EN). Four component matchers—the Qwen3-Reranker-8B reranker (a decoder trained for reranking), a rubric LLM-as-a-judge, and two bi-encoders (BGE-LoRA and MiniLM)—are scored in parallel and combined by a geometric mean of their per-job min–max-normalised scores; the fused top-60 then enters a three-round listwise tournament whose final round reorders only the top-15. The ES-ES and EN-ES chains fuse two matchers (the reranker and the BGE-M3 bi-encoder) under the same geometric-mean fusion and tournament.

Bi-encoder matchers. The EN-EN chain adds two bi-encoders. BGE-LoRA pairs BGE-large-en-v1.5 with the LoRA adapter [7] `shashu2325/resume-job-matcher-lora`, trained on roughly 8k JD–résumé pairs [14]; we score each candidate by the similarity of the pooled job and résumé embeddings (standalone development MAP $\approx$0.8300). MiniLM is the fine-tuned SentenceTransformer `anass1209/resume-job-matcher-all-MiniLM-L6-v2` [15, 16], a 22M-parameter all-MiniLM-L6-v2 with a cosine objective over $\sim$958 pairs, run on CPU (standalone development MAP $\approx$0.7620).

Geometric-mean fusion. Within each job we rescale every matcher's scores to $[0, 1]$, multiply across matchers, and take the n-th root. We selected this by sweeping every ≥ 2-component subset against reciprocal-rank fusion [17] at $k \in \{10, 60\}$, z-score averaging, and weighted min–max blending; the geometric mean dominated, with equal weighting beating learned weights. Per-job normalisation keeps the fusion robust to scale differences across matchers, so no single matcher dominates the combined ranking.

Tournament listwise reranker. The fused run seeds a listwise tournament [10] over its top-60, run by Qwen3-235B-A22B-Instruct-2507 in three elimination rounds. Round 1 splits the 60 into 10 interleaved groups of 6 and keeps the top-3 each (30 survivors); Round 2 re-groups into 5 groups of 6, keeping top-3 (15); Round 3 applies a sliding window (width 6, stride 3) over the top-15, while positions 16–60 retain their fused order. The submitted tournament forces a structured chain-of-thought JSON schema in which the model first commits to an explicit `target_title` and `target_seniority` level (`entry`, `mid`, `senior_ic`, `manager`, `director`, `vp_plus`), then emits a per-candidate `assessment` before ranking. Across directions the tournament adds roughly $+0.012$ to $+0.020$ MAP over the best fusion alone; the full prompt is given in Appendix A.2.

Per-direction configuration. The EN-EN direction fuses all four matchers (dev 4-way fusion 0.8761 MAP, tournament 0.8839). The ES-ES and EN-ES directions fuse two—the reranker and BGE-M3 [18] (`BAAI/bge-m3`, 1024-dim multilingual, standalone ES-ES development MAP $\approx$0.8330)—dropping the rubric. The ES-ES 2-way fusion reaches 0.8860 MAP, tournament 0.9011; EN-ES inherits the ES-ES configuration. An explored 3-way Spanish variant adding BGE-LoRA over a DeepSeek-V4-Flash [19] translation reached 0.9037 on dev but was not submitted.

Table 1
Task A EN-EN development MAP (10 queries): the fusion ladder from individual matchers to the submitted system. The four matchers (the Qwen3-Reranker-8B reranker, MiniLM bi-encoder, rubric LLM-as-a-judge, BGE-LoRA bi-encoder) are combined by a geometric mean of their per-job min–max-normalised scores, and the listwise tournament reorders the fused top-60. For scale, a random ranking scores 0.1018; the alternative TourRank aggregator (Section 3.4) reaches 0.8599. The last row is the submitted system.

Configuration	MAP [DEV EN-EN]
Qwen3-Reranker-8B only	0.6969
MiniLM bi-encoder	0.7620
Rubric LLM-as-a-judge	0.7791
BGE-LoRA bi-encoder	0.8300
Four-way fusion (geometric mean)	0.8761
+ **Tournament (submitted)**	**0.8839**

Table 2
Task A ES-ES development MAP (10 queries). The submitted Spanish system fuses the reranker with the BGE-M3 bi-encoder under a geometric mean and reorders with the tournament; the Spanish rubric is dropped because adding it hurt fusion (its standalone MAP, and the English-question collapse it recovers from, are shown for context). An explored three-way variant adding a BGE-LoRA encoder over a translated corpus reached 0.9037 on dev but was not submitted (see text).

Configuration	MAP [DEV ES-ES]
Rubric judge, English-language questions	0.5950
Rubric (Spanish), standalone	0.7670
BGE-M3 bi-encoder	0.8330
Qwen3-Reranker-8B alone	0.8827
Two-way fusion (geometric mean)	0.8860
+ **Tournament over fusion (submitted)**	**0.9011**

Table 3
Task A official test results (TEST). All values are on the held-out test split and are not comparable to the development MAP in Tables 1–2.

Metric	Value [TEST]
Overall AVG_MAP (EN-EN,ES-ES)	0.6532
EN-EN MAP	0.6552
ES-ES MAP	0.6512
Cross-lingual MAP (EN-ES)	0.6438
Bias-controlled AVG_RBO	0.8734

3.3. Experiments and Results

Development results. Table 1 traces the EN-EN fusion ladder on the 10-query development set: the four-way geometric-mean fusion reaches 0.8761, well above any single component, and the listwise tournament raises this to 0.8839, the best configuration we found (MRR 1.0000, P@5 1.00, NDCG $\approx$ 0.9656, P@10 0.940, P@100 0.4280).

In Spanish (Table 2) the reranker is already strong (0.8827); the submitted two-way fusion with BGE-M3 reaches 0.8860 and the tournament raises it to 0.9011.

Official test results. On the official test set (Table 3), the tournament achieves overall multilingual $\mathrm{AVG\_MAP} = 0.6532$ (EN-EN 0.6552, ES-ES 0.6512) and cross-lingual $\mathrm{MAP(EN\text{-}ES)} = 0.6438$. Its weakest metric is the bias-controlled rank-stability score $\mathrm{AVG\_RBO} = 0.8734$ (RBO EN-ES 0.8631, ES-ES 0.8837).

The development-to-test gap. The submitted system scores 0.8839 MAP on EN-EN dev but 0.6552 on test; we regard the gap as expected given the 10-query dev split's high variance and a test distribution dominated by Senior/Principal and domain-fused titles.

3.4. Error Analysis and Negative Results

Two lessons stand out. **(1) Structured-prompt commitment fixes template-clone confusion:** an early free-form prompt promoted near-duplicate résumé templates across seniority levels, and forcing an explicit `target_seniority`/`target_title` commitment before scoring eliminated the failure mode. **(2) Spanish-rubric language-match:** English-question rubrics over Spanish résumés collapse the judge (ES-ES rubric MAP 0.5950 vs. reranker 0.8827), and rewriting the rubric in Spanish recovered it (+0.172 mean MAP).

Negative results. Two listwise alternatives underperformed the submitted tournament: a more aggressive variant ($\text{TOP\_N} = 100$, deeper reordering) regressed to EN-EN 0.8207 / ES-ES 0.7948, and TourRank [11] (Borda aggregation, $\sim 3\times$ cost) reached only 0.8599 EN-EN with an identical P@100. Both confirm that once the listwise input is near-optimally ordered, further listwise sophistication is bounded by the composition and depth of that input rather than the reranker's scoring: on hard jobs $\sim 25\%$ of relevant candidates sit below the untouched rank-60 cut and are unrecoverable by reranking.

4. Task B: Job–Skill Matching with Skill Type Classification

Our Task B system is a four-stage pipeline: (i) a LoRA-fine-tuned Qwen3-Embedding retriever with LLM-expanded queries, (ii) candidate-pool expansion using JobBERT-v3, (iii) a Qwen3-Reranker-8B pointwise reranker (a decoder trained for reranking), and (iv) a Qwen3-235B select-then-rank listwise reranker over the top-300 fused candidates. The corpus is ESCO [5] skills ranked against short job titles, each labelled *core* (graded relevance 2) or *contextual* (1) [4]. Our key empirical finding is that fusing JobBERT-v3 [9] *before* the listwise stage, rather than only at score blending, raises validation graded NDCG from 0.7268 to 0.7734 (+6.4% relative): the bottleneck is the *composition* of the top-n set the listwise stage sees, not the reranker's scoring quality. On the official test set the submitted system reached graded NDCG 0.7793.

4.1. Data Analysis

The dataset has $\sim$3,011 training job–skill assignments and 304 validation queries (test: 1,139). Three data properties drove the design. **(1) Distribution shift:** training jobs have a median of 36 skills, validation jobs a median of 181 (roughly $5\times$ fan-out), and training labels are only weak predictors of validation grade (31.7% essential→core vs. 23.6% optional→core), so we treat all training positives as one class. **(2) Sparse connectivity:** 76.4% of random training-job pairs share *zero* skills, so per-skill sub-query expansion cannot recover the breadth of the relevant set. **(3) Oracle coverage:** the best single training job covers only 12.1% of a query's relevant skills, and oracle selection of 5/10/20/50 jobs covers 35.7%/51.2%/68.9%/90.4%, so direct query–corpus similarity must be the primary signal with training transfer only auxiliary. Accordingly the retriever is a query–skill direct-similarity model, training labels are binary InfoNCE supervision, and every stage widens or reorders a wide pool rather than narrowing it.

4.2. System Overview

The pipeline has four layers (Figure 2). Four per-candidate scores recur: r (Qwen3-Reranker-8B pointwise reranker), ℓ (LoRA-Qwen3 retriever), j (JobBERT-v3 direct query–skill cosine), and s (listwise select-then-rank). We blend at two points: the *pre-SR* blend reshapes the top-n set the LLM sees, and the *post-SR* blend is applied to the listwise output. The pre-SR blends are labelled *Strategy A–G* in

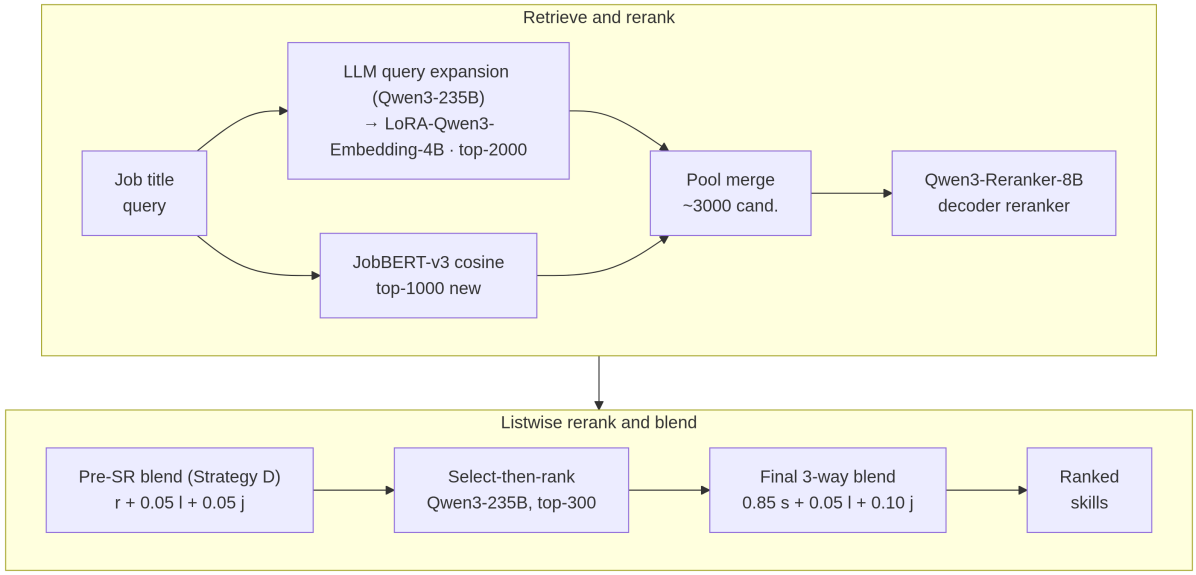

Figure 2: Submission pipeline for TalentCLEF 2026 Task B. All blends use per-query min–max normalization. r: 8B rerank score; ℓ: LoRA retriever score; j: JobBERT-direct cosine; s: select-then-rank score. Validation graded NDCG $= 0.7734$; official test graded NDCG $= 0.7793$.

Table 6; *Strategy D*, $r + 0.05\,\ell + 0.05\,j$, is submitted. All blends are per-query min–max normalized linear combinations.

4.2.1. Retriever

We fine-tune `Qwen3-Embedding-4B` [6] with LoRA [7] on the TalentCLEF training set via InfoNCE with in-batch negatives, each pair associating a job title with one of its skills. We use rank $r = 16$, $\alpha = 32$, dropout 0.05, batch size 128, learning rate 3×10^{-5}, 1 epoch. Both this fine-tuning ($\sim$1 hour) and the retriever encoding of the corpus and queries run on a single NVIDIA RTX A6000 (48 GB) that we rent on vast.ai for a few hours—the only GPU compute in the pipeline. At inference we expand each query title with a short Qwen3-235B prompt producing an ESCO-style skill description (full prompt in Appendix A.3), concatenated with the title before encoding (+4% NDCG, row 5 of Table 4). We retrieve the top 2,000 per query (recall@2000 $= 0.92$).

4.2.2. Pool expansion with JobBERT-v3

JobBERT-v3 [9] is a 1024-dim multilingual sentence encoder contrastively pre-trained on 21.1M job titles in EN/DE/ES/ZH. We encode the corpus once and add the top-1,000 query–skill cosine matches *not* in the LoRA retriever's top-2000, raising pool recall from 0.921 to 0.976.

4.2.3. Pointwise reranker

We rerank the merged $\sim$3000-candidate pool with `Qwen3-Reranker-8B` [8] via DeepInfra, instruction *"Given a job title, determine if the professional skill is relevant to this position"* (NDCG 0.74 alone at top-1000).

4.2.4. Listwise select-then-rank

The top 300 reranked candidates go to a *select-then-rank* module on `Qwen3-235B-A22B-Instruct` via DeepInfra at concurrency 30, in two phases. **Halving:** candidates are split into round-robin groups of 20; for each group the LLM is given the job title, a one-line role description, and the 20 skill names and outputs the top-10 indices, repeating until $\leq$50 remain (4–5 calls/query). **Sliding window:** survivors

Table 4
Validation binary NDCG progression. Each row is a single technique added to the previous best configuration. "MR" = max retrieved. All numbers are validation (304 queries). Note this table tracks *binary* NDCG (relevance ≥ 1) for trajectory comparability; the submitted system's headline *graded* NDCG (the primary metric) is 0.7734 (Table 5), not the 0.8051 shown here.

#	Configuration	Bin. NDCG	Δ
1	Qwen3-Embedding-0.6B zero-shot, MR= 500	0.4148	–
2	+ LoRA fine-tune on training set	0.4901	+18%
3	MR 500 → 1000	0.6007	+23%
4	+ ESCO descriptions in corpus text	0.6060	+0.9%
5	+ LLM (Qwen3-235B) query expansion	0.6302	+4.0%
6	+ Qwen3-Reranker-8B, top-1000	0.6868	+9.0%
7	Switch retriever to LoRA Qwen3-Embedding-4B	0.6925	+0.8%
8	MR 1000 → 2000, blend with retriever	0.7351	+6.2%
9	+ Listwise sliding-window LLM rerank, top-200	0.7443	+1.3%
10	+ Select-then-rank (halving + sliding window)	0.7493	+0.7%
11	+ JobBERT-v3 pool expansion + Strategy D pre-SR fusion	0.8038	+7.3%
12	+ Listwise top-n 200 → 300 (submitted)	**0.8051**	+0.2%

are re-ordered by a sliding-window listwise prompt fully ranking a window of 20, moving backwards in steps of 10 (the RankGPT recipe [10]; ∼4 calls/query). The full stage averages ∼18 calls/query at top-300; eliminated candidates are placed below the ranked set in elimination order, preserving the tail.

4.2.5. Score blending

The submitted pre-SR blend is *Strategy D*, $r + 0.05\,\ell + 0.05\,j$; the post-SR blend is $0.85\,s + 0.05\,\ell + 0.10\,j$. The select-then-rank stage emits an *ordering*, not a calibrated score, so we map the final position $\rho \in \{0, \ldots, N-1\}$ of each of the N ranked candidates to a score $s = 1 - \rho/N$ (so $\rho = 0$, the top of the list, gives $s = 1$). Candidates eliminated during selection receive scores below the lowest ranked one, in their original pointwise-reranker order, so the tail is preserved. All weights were tuned on validation.

4.3. Experiments and Results

Unless stated otherwise, all numbers are validation results (304 queries) from the official `talentclef_evaluate.py` script (graded NDCG with core= 2, contextual= 1; binary NDCG with relevance ≥ 1). Test results are in Section 4.3.4.

4.3.1. Development trajectory

Table 4 traces the development trajectory from the zero-shot baseline to the submitted system.

The largest improvement on top of the strong pipeline was the JobBERT-v3 fusion (row 11, +7.3%), the central empirical claim of this section.

4.3.2. Component ablation on the submitted pipeline

The headline improvement over the cached baseline is $\Delta_{\text{graded NDCG}} = +0.0466$ (+6.4% relative), concentrated below the top-10: the JobBERT contribution is *coverage*, surfacing relevant skills the LoRA-Qwen retriever missed entirely.

4.3.3. Where the improvement comes from: pre-SR vs. post-SR fusion

We swept seven pre-SR blend strategies (Table 6), evaluating the resulting top-200 set against gold qrels.

Strategy D puts +13 additional relevant skills per query *inside* the top-200 cutoff vs. the legacy rerank-only input (Strategy A), and equal weights on both auxiliary retrievers beat any single-retriever

Table 5
Validation results, isolating each component on top of the cached baseline (304 queries). **Bold** = best in column.

Configuration	Bin. NDCG	Graded NDCG	MAP	P@100
Cached baseline (LoRA + 8B + SR + blend)	0.7479	0.7268	0.3081	0.4406
+ JobBERT post-SR blend only (3-way)	0.7812	0.7494	0.3314	0.4689
+ Strategy D pre-SR fusion ($n = 200$)	0.8038	0.7714	0.3794	0.5167
+ Strategy D pre-SR + $n = 300$ (submitted)	**0.8051**	**0.7734**	**0.3867**	**0.5341**

Table 6
Pre-listwise blend strategies, evaluated on the resulting top-200 set (304 validation queries). r: 8B rerank, ℓ: LoRA retriever, j: JobBERT-direct.

Strategy	R@200	P@200	nDCG@200	rel. in top-200
A: r only (legacy)	0.340	0.303	0.393	60.6
B: $r + 0.10 \cdot \ell$	0.349	0.311	0.407	62.1
C: $r + 0.10 \cdot j$	0.389	0.351	0.425	70.1
D: $r + 0.05 \cdot \ell + 0.05 \cdot j$	**0.411**	**0.368**	**0.452**	**73.6**
E: $r + 0.10 \cdot$ (JB-blend $\gamma = 0.5$)	0.410	0.368	0.452	73.6
F: $r + 0.20 \cdot j$	0.383	0.346	0.418	69.1
G: $r + 0.20 \cdot \ell$	0.340	0.303	0.398	60.5

blend: switching the LLM's input ordering is worth more than re-weighting its output. On the Strategy D input, graded NDCG is higher at $n = 300$ than $n = 200$ ($0.7714 \rightarrow 0.7734$), so we retain $n = 300$.

4.3.4. Official test results

On the official held-out test set (1,139 queries), the four-stage pipeline obtained graded NDCG (primary) $= 0.7793$, binary NDCG $= 0.8123$, and MAP $= 0.3889$. The test numbers are *slightly higher* than validation on all three metrics (graded NDCG 0.7793 vs. 0.7734; binary NDCG 0.8123 vs. 0.8051; MAP 0.3889 vs. 0.3867): the system generalized marginally better on test.

4.3.5. Negative results

Several additions did not yield a deployable gain. TourRank [11] (Borda aggregation over shuffled rounds) added only $+0.0007$ binary NDCG for $\sim 3\times$ the API cost, the 235B reranker being already robust to input-ordering bias. Skill-level RRF expansion into ~ 30 sub-queries per query dropped binary NDCG $0.6302 \rightarrow 0.5207$ (-17%), as each sub-query sees only its own neighbourhood and rare skills fall out of the fused top-K. Distilling the 8B reranker into a LoRA 0.6B student [12] matched the teacher at top-200 but lost the top-1000 tail. AgentIR-style [13] synthetic-reasoning training helped the 0.6B retriever ($+3.3\%$ NDCG) but *hurt* the 4B (-7.5% recall), which is already at the data ceiling. Scaling the retriever 4B$\rightarrow$8B added only $+0.6\%$ recall, and ESCO corpus enrichment ($+0.9\%$ alone) was subsumed by the reranker, while co-occurrence/hierarchy boosts hurt (e.g. $\gamma = 0.3$: -34%). Per-query reasoning expansion added a negligible $+0.1\%$ but is essentially free (one cached LLM call per query), so we kept it.

4.4. Discussion

Pre-listwise fusion beats post-blending because auxiliary signals re-applied at the listwise *output* can nudge present items but cannot recover missing ones, so in pipelines with a hard top-n cut, signal injection has greater marginal value the closer it is to the cut. The retained final-blend weights $0.85 \cdot s + 0.05 \cdot \ell + 0.10 \cdot j$ give JobBERT more residual weight than the LoRA retriever, whose signal the pipeline has largely absorbed—suggesting a JobBERT-v3 fine-tuned on the training set could replace it.

4.5. Cost and Reproducibility

End-to-end test inference for 1,139 queries takes $\sim$3 hours, dominated by Qwen3-235B select-then-rank ($\sim$110 min, $\sim$33k requests at concurrency 30); all heavy computation runs on the DeepInfra API apart from LoRA training and the retriever/JobBERT encoding, which run on a single NVIDIA RTX A6000 rented on vast.ai for a few hours. The pipeline is reproducible from one driver script and $\sim$10 component scripts, configured via the `DEEPINFRA_API_KEY` variable and the public TalentCLEF 2026 Task B corpus (Zenodo record 19652670).

5. Conclusion

Both bipboopbipboop submissions to TalentCLEF 2026 instantiate a common retrieval $\rightarrow$ pointwise-reranking $\rightarrow$ listwise-reranking pipeline. The unifying lesson is that when the final LLM stage sees only the top-n of its input, the binding constraint is the *composition* of that top-n, not the listwise model's scoring: on Task B, fusing JobBERT-v3 before the listwise stage raises validation graded NDCG from 0.7268 to 0.7734 ($+6.4\%$ relative). On the official test sets, Task A reached AVG_MAP 0.6532 and cross-lingual MAP 0.6438, and Task B reached graded NDCG 0.7793, binary NDCG 0.8123, and MAP 0.3889.

Acknowledgments

We thank the TalentCLEF 2026 organisers for running the lab and for providing the datasets and evaluation infrastructure, and the maintainers of the open-source models and tooling used in this work.

Declaration on Generative AI

During the preparation of this work, the authors used an AI assistant in order to: Drafting content, Paraphrase and reword, and Formatting (LaTeX). After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

A. Prompts

A.1. Task A: sample rubric question

The per-job rubric is generated by Qwen3-235B; each question is answered Yes (1) / Partial (0.5) / No (0) from the résumé alone. A representative category and two of its questions (the placeholders < . . . > are filled per job):

```
1. Each question MUST be answerable with **Yes (1) / Partial (0.5) / No (0)** by reading only the
candidate's resume.
...
{
  "job_title": "<extracted job title>",
  "total_questions": <integer>,
  "categories": [
    {
      "id": 1,
      "name": "Hard Skills & Technical Competencies",
      "weight": 0.20,
      "questions": [
        {"id": "Q1", "text": "Does the candidate list <specific skill> in their skills section or
demonstrate it through work experience?"},
        {"id": "Q2", "text": "Does the candidate provide a concrete accomplishment demonstrating
proficiency in <specific skill>?"}
      ]
    }
```

```
  ]
}
```

A.2. Task A: listwise tournament prompt

```
You are an expert recruiter ranking candidates for a specific job posting.

RANKING CRITERIA - in priority order:
1. TITLE FAMILY: the candidate's most recent role must be in the same job family as the posting. A '
Paralegal' posting wants Paralegals or Legal Assistants - NOT Compliance Directors, even if their
resume mentions privacy/legal keywords as part of oversight scope.
2. SENIORITY BRACKET: match individual contributor vs manager vs director vs VP+. A 'Senior <role>'
posting wants senior individual contributors of that role, NOT executives in adjacent domains.
3. HANDS-ON RESPONSIBILITIES: reward verbs/tasks that match the posting's core work, not high-level
oversight language that merely mentions the same topics.
4. SKILLS / DOMAIN / KEYWORDS: tie-breakers only - never promote across a title or seniority gap on
the strength of keyword overlap alone.

WATCH OUT FOR:
- Director/VP/Chief resumes that name the posting's keywords as part of oversight duties.
- Near-duplicate template resumes - count the cluster as a single signal, not three.
- Generic management language when the posting requires hands-on specialist work.

Return ONLY a valid JSON object - no explanation outside JSON, no markdown.

[Per-query user message, candidates omitted:]

## TASK
Step 1 - Identify the posting's TARGET TITLE and SENIORITY BRACKET.
Step 2 - For each of the N candidates, name their most recent title and seniority bracket. Stating the
 title forces it into your decision.
Step 3 - Select the top K. Do NOT promote candidates across a title-family or seniority gap on keyword
 overlap alone.

Return JSON exactly in this shape (and nothing else):
{
  "target_title": "<role from posting>",
  "target_seniority": "<entry|mid|senior_ic|manager|director|vp_plus>",
  "assessment": [
    {"id": "<id>", "title": "<recent title>", "seniority": "<bracket>", "family_match": true}
    // one entry per candidate
  ],
  "top_K": ["<id_best>", "<id_second>", ...]  // length K, ordered best->worst
}
```

A.3. Task B: query-expansion prompt

```
You are an expert in job markets and professional skills taxonomy (ESCO framework).
You will receive a job title from a real-world job posting. Do two things:

1. **CLEANED TITLE**: Normalize the job title:
   - Expand abbreviations (sw -> software, gl -> general ledger, sdet -> software development engineer
 in test, crm -> customer relationship management, iam -> identity and access management)
   - Keep seniority levels (senior, junior, lead, associate, intern, trainee, apprentice, ii, iii) -
they affect required skills
   - For compound roles with "/", keep all parts separated by comma
   - Keep it concise, just the normalized role name(s)

2. **SKILLS**: List ALL professional skills, knowledge, competences, abilities, and other
characteristics required or desired for this position. Be extremely exhaustive. Cover every category:
   - Hard/technical skills (programming languages, frameworks, tools, methodologies)
   - Domain knowledge (industry-specific knowledge areas, regulations, standards)
   - Soft skills and interpersonal competences (communication, leadership, teamwork)
   - Cognitive abilities (analytical thinking, problem solving, creativity)
   - Tools, software, and technologies
   - Certifications and qualifications
   - Work style and personal attributes (attention to detail, adaptability, resilience)
   For each skill provide:
   - "title": a short skill name
   - "description": a one-sentence description of the skill in context of this role
```

```
   - "type": either "essential" (core requirement, the role cannot be performed without it) or "
optional" (nice-to-have, beneficial but not strictly required)

Respond in JSON format exactly like this:
{"cleaned": "normalized job title", "skills": [{"title": "skill name", "description": "one-sentence
description", "type": "essential"}, ...]}
```

References

[1] L. Gasco, H. Fabregat, L. García-Sardiña, P. Estrella, D. Deniz, A. Rodrigo, R. Zbib, Overview of the talentclef 2025: Skill and job title intelligence for human capital management, in: International Conference of the Cross-Language Evaluation Forum for European Languages, 2025, pp. 464–485.

[2] L. Gasco, H. Fabregat, L. García-Sardiña, P. Estrella, W. Veys, C. P. Carrino, M. De Lange, D. Deniz Cerpa, A. Rodrigo, J.-J. Decorte, R. Zbib, Overview of the talentclef 2026: Skill and job title intelligence for human capital management, in: International Conference of the Cross-Language Evaluation Forum for European Languages, 2026.

[3] H. Fabregat, L. García-Sardiña, P. Estrella, L. Gasco, C. P. Carrino, D. Deniz Cerpa, A. Rodrigo, R. Zbib, Overview of talentclef 2026: Task a — contextualized job-person matching, 2026.

[4] W. Veys, L. Gasco, M. De Lange, J.-J. Decorte, Overview of talentclef 2026: Task b — job-skill matching with skill type classification, 2026.

[5] European Commission, ESCO — european skills, competences, qualifications and occupations, https://esco.ec.europa.eu/, 2024. Accessed 2026.

[6] Qwen Team, Qwen3-embedding: Multilingual text embedding models, https://huggingface.co/Qwen/Qwen3-Embedding-0.6B, 2025. Accessed 2026.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: International Conference on Learning Representations (ICLR), 2022.

[8] Qwen Team, Qwen3-reranker: Cross-encoder reranking models, https://huggingface.co/Qwen/Qwen3-Reranker-8B, 2025. Accessed 2026.

[9] TechWolf, JobBERT-v3: Multilingual job-title sentence encoder, https://huggingface.co/TechWolf/JobBERT-v3, 2025. Accessed 2026.

[10] W. Sun, L. Yan, X. Ma, S. Wang, P. Ren, Z. Chen, D. Yin, Z. Ren, 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), 2023, pp. 14918–14937.

[11] Y. Chen, Q. Liu, Y. Zhang, W. Sun, D. Shi, J. Mao, D. Yin, TourRank: Utilizing large language models for documents ranking with a tournament-inspired strategy, in: Proceedings of the ACM Web Conference (WWW), 2025. https://arxiv.org/abs/2406.11678.

[12] ModelScope, ms-swift: Scalable lightweight infrastructure for fine-tuning, https://github.com/modelscope/ms-swift, 2025. Accessed 2026.

[13] Z. Chen, X. Ma, S. Zhuang, J. Lin, A. Asai, V. Zhong, AgentIR: Reasoning-aware retrieval for deep research agents, CoRR abs/2603.04384 (2026). URL: https://doi.org/10.48550/arXiv.2603.04384. doi:10.48550/ARXIV.2603.04384. arXiv:2603.04384.

[14] Shashu, resume-job-matcher-lora: A LoRA adapter on BGE-large-en-v1.5 for résumé–job matching, https://huggingface.co/shashu2325/resume-job-matcher-lora, 2024. Accessed 2026.

[15] Anass, resume-job-matcher-all-MiniLM-L6-v2: A fine-tuned SentenceTransformer for résumé–job matching, https://huggingface.co/anass1209/resume-job-matcher-all-MiniLM-L6-v2, 2024. Accessed 2026.

[16] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3982–3992.

[17] G. V. Cormack, C. L. A. Clarke, S. Büttcher, Reciprocal rank fusion outperforms condorcet and individual rank learning methods, in: Proceedings of the 32nd International ACM SIGIR Conference

on Research and Development in Information Retrieval (SIGIR), 2009, pp. 758–759.

[18] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, Z. Liu, BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, arXiv preprint arXiv:2402.03216 (2024). https://huggingface.co/BAAI/bge-m3.

[19] DeepSeek-AI, DeepSeek-V4-Flash: Fast instruction-tuned language model, https://huggingface.co/deepseek-ai, 2025. Served via DeepInfra.