# Semantic Enrichment for Resume Ranking and Skill Retrieval at TalentCLEF 2026

Working note for the TalentCLEF Lab at CLEF 2026

Sangkeun Kim[1], Minseong Choi[1] and Seungho Lee[2,*]

[1]CLASSUM, 27, Teheran-ro 2-gil, Gangnam-gu, Seoul, Republic of Korea
[2]Lafit Inc., 101-J18, 154-29, Imbangul-daero, Gwangsan-gu, Gwangju, 62238, Republic of Korea

**Abstract**

We present our system for TalentCLEF 2026 shared tasks A and B. The central idea is semantic enrichment before retrieval. Instead of scoring the original text fields directly, the system constructs task specific representations that make roles, work activities, skill requirements, and skill meanings easier to compare. In Task A, job descriptions and resumes are enriched with structured views that expose role compatibility, activity overlap, seniority, domain, and skill evidence. In Task B, job titles are expanded into skill oriented profiles, while ESCO skills are represented with metadata enriched descriptions. The enriched representations are used with dense retrieval models, large language model based scoring components, score fusion, and limited reranking. The same design principle led to competitive results across the two tasks, despite their different inputs and evaluation metrics. Our submissions obtained 0.7140 average MAP in Task A, 0.7044 MAP in the Task A English to Spanish setting, and 0.8068 graded NDCG in Task B.

**Keywords**

Skill and job intelligence, Semantic enrichment, Dense retrieval, Contrastive learning

## 1. Introduction

Job and skill matching systems often compare texts that describe different levels of work. A job description states responsibilities and requirements. A resume provides evidence of prior work. A job title compresses a role into a short phrase. A skill entry describes a capability that may be required, useful, or only indirectly related to the role. These fields are connected by activities, seniority, tools, domain context, and skill requirements, but these links are often only partially visible in the original text.

This paper describes our system for TalentCLEF 2026 shared tasks A and B [1]. Task A, Contextualized Job-Person Matching, ranks candidate resumes for job descriptions in English and Spanish, including monolingual, cross lingual, and bias controlled evaluation settings [2]. Task B, Job-Skill Matching with Skill Type Classification, retrieves ESCO skills for job titles and evaluates graded relevance between the title and the skill [3, 4]. The two tasks differ in input length, candidate space, and supervision, but both require matching beyond direct surface overlap.

Our system is built around semantic enrichment before scoring. The goal is to convert sparse or heterogeneous input fields into task specific representations that expose the evidence needed by retrieval and scoring models. This design is related to retrieval methods that use representation expansion to reduce query document mismatch [5, 6], and to dense retrieval methods that use learned encoders for scalable semantic matching [7, 8, 9]. It is also consistent with prior TalentCLEF systems, which combined domain specific representations, multilingual encoders, contrastive learning, large language model augmented data, and reranking rather than relying on a single generic model [10, 11, 12, 13].

*CLEF 2026 Working Notes, 21 to 24 September 2026, Jena, Germany*
*Corresponding author.
windy@classum.com (S. Kim); alias@classum.com (M. Choi); aiden@lafit.dev (S. Lee)
https://classum.com (S. Kim); https://lafit.dev (S. Lee)
0000-0001-9163-7100 (S. Kim); 0009-0005-9255-7109 (M. Choi); 0009-0006-0151-8292 (S. Lee)

The resulting systems combine enriched representations with complementary dense, structured, and large language model based scores. Reranking is used only as a constrained refinement step. We instantiate this design for both resume ranking and ESCO skill retrieval, and report competitive official results in both tasks.

## 2. Setup

TalentCLEF 2026 shared task A, Contextualized Job-Person Matching, evaluates resume ranking for job descriptions. The released data covers English and Spanish, and the official evaluation includes monolingual, cross lingual, and bias controlled settings. Mean Average Precision is the main ranking metric [1, 2].

TalentCLEF 2026 shared task B, Job-Skill Matching with Skill Type Classification, evaluates skill retrieval for job titles. Systems retrieve ESCO skills and rank them according to graded relevance. The primary metric is graded NDCG, with binary NDCG, MAP, and MRR also reported. The validation set contains 304 job title queries and 9,052 candidate skills. The official test run required ranked lists for 1,139 job title queries [1, 3].

ESCO is a multilingual classification of European skills, competences, qualifications, and occupations developed by the European Commission Directorate General for Employment, Social Affairs and Inclusion in collaboration with stakeholders. It describes, identifies, and classifies occupations and skills for labour market and education applications. The dataset is publicly downloadable free of charge in 28 ESCO languages and in multiple formats, including RDF, TTL, ODS, CSV, XML, and JSON-LD. Its skill entries include labels, descriptions, reuse levels, broader concepts, and occupation links, which make it useful as a structured source for skill and job intelligence [4, 14].

## 3. Method

Both tasks share a single design principle, semantic enrichment before scoring: sparse or heterogeneous inputs are converted into task specific comparison views that expose the evidence retrieval and scoring models need. Released query and candidate records are first converted into these enriched views, which are scored by complementary dense, structured, and large language model based components. Final rankings are produced by per query score normalization, score fusion, and constrained reranking.

### 3.1. Task A

For Task A, our system aggregates evidence over enriched job and resume views. The official development judgments were used to extract general matching principles rather than to train a task specific encoder. We summarized the judgments into rules about transferable work activities, seniority mismatch, domain proximity, and rejection criteria. These rules were used as instructions for selected large language model based scoring and reranking components.

We also constructed an internal diagnostic validation surface from released text. An internal extraction pipeline produced skill and task profiles for job descriptions and resumes, and compared pairs through profile overlap and role proximity. Concretely, an instruction tuned model was prompted with an expert HR analyst rubric that separates skills (“can do” capabilities) from tasks (“do” activities), tags each skill with a character type and a role specific versus transferable label, and returns structured JSON; the full view construction and the resulting profile overlap signal are described in Appendix A. The matching principles used by the listwise and rubric components were distilled from the development judgments, as detailed in Appendix B, with example principles in Table 9. This surface was used together with the official development split for conservative ensemble selection. For each candidate subset, we evaluated both development MAP and diagnostic MAP, and selected subsets by the lower of the two scores. The scoring pool contained heterogeneous components rather than a single retriever. We use short

component names in this section; Table 8 and the reproducibility notes resolve every short name to its exact public checkpoint or hosted service identifier.

The structured embedding components operate on enriched views rather than raw text. A job description or resume is first parsed into a fixed schema (title, seniority, years of experience, education, domain, sub-specialization, certifications, and a responsibilities or career summary), merged with the skill and task extraction described above, and serialized into labeled `Field: value` lines that are embedded with a retrieval query or retrieval document task type. Table 1 shows a worked example for a development pair. The enrichment exposes evidence that surface text hides: the job and the resume share generic "engineering / data / systems" vocabulary, but the explicit `Domain` and `Role-specific skills` fields make the Healthcare / Data-Analytics versus Electrical-Engineering mismatch directly visible to the embedding model rather than rewarding incidental lexical overlap.

**Table 1**
Example of representation enrichment for Task A (structured view), built from a development job description and resume.

| Field | Text |
|---|---|
| Raw job description (excerpt) | `Healthcare Data Analyst.` We are seeking a Healthcare Data Analyst to collect, clean, analyze, and interpret data from multiple sources within the healthcare domain, develop and maintain data models and pipelines, build reporting dashboards, … [followed by a flat list of surface skill phrases]. |
| Enriched job-description view | Title: Healthcare Data Analyst. Seniority: mid. Domain: Healthcare / Data Analytics. Specialization: Clinical and Operational Analytics. Summary: collecting, cleaning, and analyzing healthcare data to develop data models, pipelines, and visualizations that support clinical and administrative decision-making. Role-specific skills: Healthcare Industry Knowledge; Data Quality; Data Modeling; Data Security; Data Pipeline Development; Standards Compliance; Extract, Transform, Load (ETL). Transferable skills: Data Analysis; Data Visualization; Business Intelligence; Statistics; Analytics; Business Process Analysis. Key tasks: collect, clean, and harmonize healthcare data from diverse sources; develop data models, pipelines, and data products; create dashboards and reports; monitor data quality and governance; conduct statistical analyses to inform clinical and operational decisions. |
| Enriched resume view | Title: Electrical Engineer. Seniority: mid. Years of Experience: 6 years. Education: BSc Electrical Engineering. Domain: Electrical Engineering. Specialization: Industrial Automation & Power Distribution. Certifications: Professional Engineer (PE) License. Summary: licensed PE with 6 years in electrical design, industrial automation, and power distribution. Role-specific skills: Industrial automation; Power distribution; Variable frequency drives; National Electrical Code (NEC); NFPA standards; AutoCAD; Control systems; Instrumentation; Electrical testing. Transferable skills: Sustainability. Key tasks: lead electrical design and engineering activities; implement motor control systems; conduct commissioning activities; prepare technical documentation; provide troubleshooting support. |

- **Dense embedding components.** We evaluated multilingual and cross lingual embedding backbones, including JobBERT-v3, KaLM12B, Octen8B, Nemotron8B, ZEmbed1-4B, and Harrier27B variants. These components provided broad semantic matching signals under the standard bi encoder retrieval paradigm [7, 9, 11, 15, 16]. Not all backbones entered the final ensembles: JobBERT-v3 and the Harrier27B and Octen8B variants were used as candidate evidence sources during selection but were not retained in any submitted subset (Table 2), where stronger general purpose encoders dominated.
- **Structured embedding components.** We used Gemini001 on extracted job and resume views. The views included title only, summary, task list, and GPT distilled structured profiles containing skills, tasks, and responsibilities. Retrieval query and retrieval document task types were used for the two sides of the comparison [17].
- **Listwise scoring components.** We used GPT5.5 and ClaudeOpus4.6 for listwise scoring over selected candidate sets. These components were used to judge compatibility when the match

depended on implicit transfer between work histories and job requirements rather than direct lexical or embedding similarity.

- **Rubric scoring components.** We used Gemini3 Flash to produce compact rubric scores for title, seniority, function, and domain compatibility. These scores were coarse by design and served as stabilizing evidence for the ensemble.
- **Reranking components.** We used Cohere Rerank4 and Zerank2 for candidate reranking. CohereRerank4 was used as a hosted reranking component, while Zerank2 was used as a public reranker checkpoint. Rerankers were applied only to selected candidate sets or local score plateaus, so they refined the fused retrieval ranking rather than replacing it.

Each component produced a score for every job and resume pair. Scores were normalized independently for each query with min max scaling. The final score was a sum over the selected component subset for each language setting,

$$S_\ell(q,d) = \sum_{m \in \mathcal{M}_\ell} w_m \tilde{s}_m(q,d), \qquad w_m \in \{0,1\},$$

where $\ell$ denotes the language setting, $\mathcal{M}_\ell$ is the selected component subset, and $\tilde{s}_m$ is the normalized score from component $m$. The fusion deliberately uses no continuously tuned weights: every selected component enters with equal weight, so the only degree of freedom is binary inclusion ($w_m = 1$) or exclusion ($w_m = 0$). We chose this after a controlled comparison in which linear, squared, cubic, exponential, shifted, and rank based weighting schemes derived from standalone development MAP were all outperformed by equal weighting on the development set (0.9339 for equal weighting versus lower scores for every weighted scheme), consistent with equal weighting preserving ensemble diversity. The included subset $\mathcal{M}_\ell$ was selected per language setting by exhaustive search over all non-empty subsets of the candidate pool, scoring each subset by both official development MAP and diagnostic MAP and ranking by the lower of the two to avoid overfitting either label source. We selected different subsets for English to English, Spanish to Spanish, and English to Spanish because the strongest signals were not identical across settings. Table 2 summarizes the selected evidence components for each language setting.

**Table 2**
Task A selected evidence components by language setting.

| Setting | Selected evidence components |
|---|---|
| English to English | GPT5.5 List, Nemotron8B Dense, ClaudeOpus4.6 List, Gemini3 Flash Title Rubric, Gemini001 Title |
| Spanish to Spanish | GPT5.5 List, KaLM12B Dense, ClaudeOpus4.6 List, Gemini3 Flash Title Rubric, Gemini001 Task, Gemini001 Profile, Gemini001 Title |
| English to Spanish | ZEmbed1-4B Dense, ClaudeOpus4.6 List, Gemini3 Flash Title Rubric, Gemini001 Title, Gemini001 Task |

Two constrained post processing steps were applied after score fusion. First, scores were smoothed over canonical role groups in the released corpus. Candidate resumes corresponding to the same occupational role were averaged and the group score was broadcast back to the variants. This reduced sensitivity to incidental persona wording while keeping the smoothing operation query local. Second, we detected score plateaus in the sorted ensemble scores and applied Zerank2 only within each plateau. The order between plateaus was preserved, so reranking acted as local refinement rather than a replacement for the fused retrieval ranking.

## 3.2. Task B

For Task B, our system scores skills using enriched representations of job titles and skill entries. The main mismatch is asymmetric underspecification. A job title is often too short to expose the skills

required by the role, while a skill alias may omit the description, hierarchy, and context needed to judge graded relevance. We therefore expanded job titles into skill oriented profiles and rendered skills as concept cards using labels, descriptions, reuse levels, broader concepts, and occupation links, as illustrated in Table 3; the job title decomposition prompt is given in Appendix C.

**Table 3**
Example of representation enrichment for Task B.

| Field | Text |
|---|---|
| Raw job title | `logistics business analyst` |
| Enriched job-title view | Job: logistics business analyst. Task facets: analyze logistics and supply chain processes; gather and document business requirements; translate business needs into process maps and functional specifications; monitor logistics KPIs such as on-time delivery, inventory turnover, and fill rate; use data from ERP, WMS, TMS, and spreadsheets. Compact skill cards: business process modeling; supply chain management; inventory management principles; warehouse management systems; SQL; data visualization; quantitative reasoning; systems thinking; cross-functional collaboration; stakeholder communication. |
| Raw skill alias | `use methods of logistical data analysis` |
| Enriched skill-concept view | Skill: use methods of logistical data analysis. Aliases: interpret logistical data; analyse logistical data; interpret data on elements of logistics; use logistical data analysis methods. Description: read and interpret supply chain and transportation data; analyse the reliability and availability of findings using data mining, data modelling, and cost-benefit analysis. Skill type: skill. Reuse level: cross-sector skills and competences. Broader concept: analysing business operations. Linked occupations: logistics analyst; logistics engineer; civil engineer; application engineer; transport engineer. |

Training used the official occupation skill supervision converted to the same profile and concept card surface. Core skills were sampled with higher weight than contextual skills. We also added a small supplemental set assembled from publicly accessible job postings collected from the career pages of a range of companies. Each posting was processed by an instruction tuned model that extracted the role's tasks and required skills, and the result was normalized into ESCO style title skill alignments, that is, job titles paired with skill phrases mapped onto the same skill oriented profile and concept card surface used for the official data. This supplemental set contained on the order of 900 job titles and about 12,000 extracted title skill pairs. Because it is derived from third party postings it is not redistributable, and it was used only as additional semantic coverage; its effect was positive but marginal, so all reproducible results rely on the official ESCO supervision and the generated concept card representations rather than on this supplemental set.

The dense retrieval pool used several embedding backbones instead of a single encoder. The final pool combined ZEmbed1-4B, KaLM12B, Qwen3-8B, and NVEmbed3B, while Harrier27B LoRA variants were trained but not selected for the final ensemble [18, 15, 19]. Exact checkpoints are listed in the reproducibility notes.

Training was staged. We first fine tuned a ZEmbed1-4B anchor with standard in batch contrastive loss on the enriched profile and concept card surface. The other selected backbones were fully fine tuned with a guide aware contrastive recipe based on GISTEmbed and its memory efficient Sentence Transformers implementation, CachedGISTEmbedLoss [20, 21]. GISTEmbed is a variant of in batch contrastive learning in which a separate, frozen guide embedding model scores the in batch candidates and filters out likely false negatives before the contrastive loss is computed, so that semantically related items are not penalized as negatives. Training used sequence length 512, two epochs, balanced multi positive sampling with two core and two contextual positives per query, and effective global batch sizes in the 96 to 192 range. ESCO skills form a dense candidate space, where many in batch negatives can be related skills rather than true negatives. A frozen guide model was therefore used to mask candidates whose guide score was within a relative margin of 0.05 from the best in batch candidate for the same query. The student was trained with temperature 0.02 contrastive loss on the remaining tuples. We refer to this guide filtered recipe as *guide masked fine tuning* in Table 7.

Each fine tuned model produced a full depth dense run. The dense runs were combined with validation selected convex weights. The fused run was then reranked at top 512 depth using Zerank2, and the final score blended dense and reranker scores with validation selected weight $\alpha = 0.5$ [22, 23].

## 4. Results and Discussion

Table 4 reports the official results of the submitted systems. For Task A, the table separates the overall multilingual, cross lingual, and bias controlled leaderboard views. For Task B, it reports both the graded and binary evaluation views.

**Table 4**
Official TalentCLEF 2026 results.

| Task | View | Primary metric | Additional metrics |
|---|---|---|---|
| Task A | Overall multilingual | AVG_MAP 0.7140 | MAP(en-en) 0.7122 |
| | | | MAP(es-es) 0.7157 |
| Task A | Cross lingual | MAP(en-es) 0.7044 | |
| Task A | Bias controlled | AVG_RBO 0.9891 | RBO(en-es) 0.9892 |
| | | | RBO(es-es) 0.9890 |
| | | | MAP(es-es) 0.7231 |
| Task B | Graded | NDCG_graded 0.8068 | MAP_graded 0.4197 |
| Task B | Binary | NDCG_binary 0.8340 | MAP_binary 0.4197 |

The official shared task leaderboard was not available during our development window, so we cannot report a direct comparison against the organizer baseline or the participant median from our own records; where such figures are published in the task overview, they provide the appropriate external reference for the absolute numbers in Table 4. As an internal reference point, a single strong dense encoder reaches development MAP in the low 0.90s on English to English (Table 5), against which the fusion, distillation, and smoothing stages add the further gains traced in Table 6.

The validation evidence played different roles in the two tasks. In Task A, several strong off the shelf components quickly reached high development performance, so we treated them as candidate evidence sources and focused on selecting stable combinations. The main validation objective was not to establish fine grained causal claims for each component, but to identify run components that were consistent under both official development judgments and extracted diagnostic views. Table 5 reports representative component development results used in this selection process.

**Table 5**
Representative Task A component development results.

| Component | Scoring method | Dev MAP |
|---|---|---|
| English to English | | |
| ClaudeOpus4.6 | listwise rerank, full text | 0.9127 |
| Nemotron8B | dense embedding, full text | 0.9057 |
| Gemini3 Flash | four dimensional rubric, title axis | 0.8923 |
| GPT5.5 | listwise rerank, full text | 0.8881 |
| Gemini001 | dense embedding, title view | 0.8600 |
| Spanish to Spanish | | |
| Gemini001 | dense embedding, GPT distilled profile | 0.9086 |
| Gemini001 | dense embedding, task schema view | 0.9040 |
| KaLM12B | dense embedding, full text | 0.9034 |
| ClaudeOpus4.6 | listwise rerank, full text | 0.8927 |
| Gemini3 Flash | four dimensional rubric, title axis | 0.8901 |
| GPT5.5 | listwise rerank, full text | 0.8863 |
| Gemini001 | dense embedding, title view | 0.7163 |

Beyond the standalone components, Table 6 traces the contribution of each pipeline stage. Here "diagnostic MAP" is computed on a pseudo-labeled diagnostic surface that we use only for stability checking, reported alongside the official development MAP, which is available for English to English only because the released development judgments cover English. Moving from the initial fusion baseline to the robustness selected ensemble is the largest single gain on the diagnostic surface; adding the distilled abstract listwise component is near neutral on the diagnostic surface but positive on official development MAP; and role level smoothing adds a further consistent gain. We read this as evidence that the ensemble selection and the smoothing step drive most of the robust improvement, while distillation contributes a small, development verified refinement. Because role level smoothing uses only the label free role-variant structure of the released candidate corpus, its effect is not measurable on the development split, where the rankings before and after smoothing coincide.

**Table 6**
Task A stage ablation. Diagnostic MAP is the pseudo-labeled diagnostic surface (all three settings); development MAP uses the official development judgments (English to English only).

| | Diagnostic MAP | | | Dev MAP |
|---|---|---|---|---|
| Stage | en-en | es-es | en-es | en-en |
| Initial fusion baseline | 0.8589 | 0.8176 | 0.8373 | – |
| + robustness selected ensemble | 0.8828 | 0.8334 | 0.8625 | 0.9331 |
| + abstract listwise distillation | 0.8762 | 0.8333 | 0.8625 | 0.9368 |
| + role level smoothing | **0.8834** | **0.8439** | **0.8703** | 0.9368 |

In Task B, the validation set allowed direct comparison of representation choices, dense encoders, fusion, and reranking. With `Gemini001`, raw title and skill alias retrieval achieved 0.6745 graded NDCG. Replacing skill aliases with ESCO concept cards increased the score to 0.7094, and pairing generated job skill profiles with concept cards increased it to 0.7555. These representation pilots motivated the enriched surface used for later fine tuning and fusion.

Table 7 reports the Task B validation results for the final dense encoders and the fused reranked system. The gain from fusion and reranking was larger than the spread among the strongest individual dense models, which supports combining complementary dense encoders rather than selecting a single encoder. All Task B fine tuning experiments in Table 7 were run with a fixed seed and a matched training protocol. The reported fusion gain should therefore be interpreted as a controlled comparison rather than a seed averaged estimate, while multi seed evaluation that reports the distribution over runs remains a valuable robustness analysis for future work.

**Table 7**
Task B validation results.

| Base model | Training method | gNDCG | bNDCG | MAP |
|---|---|---|---|---|
| Qwen3-8B | guide masked full fine tuning | 0.7792 | 0.8097 | 0.3654 |
| NVEmbed3B | guide masked full fine tuning | 0.7824 | 0.8147 | 0.3752 |
| ZEmbed1-4B | anchor contrastive fine tuning | 0.7887 | 0.8171 | 0.3799 |
| KaLM12B | guide masked full fine tuning | 0.7894 | 0.8201 | 0.3880 |
| Four encoder fusion with top 512 Zerank2 score blend | | 0.8078 | 0.8353 | 0.4197 |

The common result across the two tasks is that semantic enrichment improved the scoring surface before ranking optimization. In Task A, candidate suitability was not reducible to a single resume embedding. Role identity, activity overlap, seniority, domain transfer, and language consistency carried distinct evidence. In Task B, job titles and skill aliases were too short to expose the distinction between core and contextual skills. Expanding both sides into a shared skill oriented format made the retrieval problem easier for embedding models.

The two tasks required different evidence standards. Task A was best treated as a stable ensemble selection problem over strong heterogeneous components. Task B allowed more direct validation driven

optimization after the enriched representation surface had been fixed. This explains why we report Task A at the run configuration level and Task B with more detailed validation comparisons.

Although the submitted systems achieved competitive results in both tasks, they should be interpreted as shared task retrieval systems rather than deployable hiring decision tools. Generated and extracted job and skill views may encode model assumptions. Role level smoothing can reduce corpus variant noise but does not establish fairness in real hiring. Provider managed LLM and reranking calls reduce external auditability, even when other components are based on public checkpoints. The results should therefore be read as evidence for a retrieval methodology under controlled shared task conditions, not as a recommendation to automate hiring decisions.

## 5. Reproducibility Notes

All Task A diagnostic and representation building steps were derived from released job description and resume text fields. The diagnostic views were used to compare system behavior and select stable ensemble variants, not as supervised training labels. Task B used official occupation skill supervision, generated profile and concept card representations, and a supplemental set of extracted occupation skill pairs. We therefore separate official supervision, generated or extracted representations, and supplemental internal data in the description above.

Most open weight embedding and reranking components, together with the contrastive training recipe, can be reproduced from public checkpoints, papers, and library documentation. Table 8 maps the component names used above to public checkpoint identifiers [24, 25, 26, 27, 28, 29, 30, 31, 23].

**Table 8**
Model identifiers documented for reproducibility.

| Component name | Public checkpoint |
|---|---|
| JobBERT-v3 | `TechWolf/JobBERT-v3` |
| KaLM12B | `tencent/KaLM-Embedding-Gemma3-12B-2511` |
| Octen8B | `Octen/Octen-Embedding-8B` |
| Nemotron8B | `nvidia/llama-embed-nemotron-8b` |
| ZEmbed1-4B | `zeroentropy/zembed-1-embedding` |
| Harrier27B | `microsoft/harrier-oss-v1-27b` |
| Qwen3-8B | `Qwen/Qwen3-Embedding-8B` |
| NVEmbed3B | `nvidia/llama-nv-embed-reasoning-3b` |
| Zerank2 | `zeroentropy/zerank-2-reranker` |

Hosted and provider managed components require a different reproducibility statement. GPT5.5 and ClaudeOpus4.6 were used for listwise scoring, Gemini001 for structured embeddings, Gemini3 Flash for rubric scoring, and CohereRerank4 for hosted reranking. Their service identifiers were `gpt-5.5`, `claude-opus-4-6`, `gemini-embedding-001`, `gemini-3-flash-preview`, and `rerank-v4.0` [32, 33, 17, 34, 35]. Exact cached API outputs and provider side snapshots are not released. The paper therefore supports checkpoint level reproduction for public models and procedure level reproduction for hosted calls, but does not claim bitwise reproduction of the submitted rankings.

## 6. Conclusion

We presented our systems for TalentCLEF 2026 Task A, Contextualized Job-Person Matching, and Task B, Job-Skill Matching with Skill Type Classification. Both systems used semantic enrichment before retrieval. Task A combined extracted job and resume views through equal weighted fusion of per query normalized scores over a robustly selected component subset. Task B aligned job title profiles and ESCO concept cards through contrastive fine tuning, dense fusion, and top depth reranking.

The results support semantic enrichment as a practical design pattern for resume ranking and skill retrieval. Enriched representations made heterogeneous inputs easier to compare, made complementary

scoring signals easier to combine, and kept reranking limited to local refinement. Future work should make this pattern more reproducible and easier to audit by reducing reliance on provider managed calls, formalizing diagnostic validation surfaces for weakly supervised ranking tasks, and evaluating whether role level smoothing improves robustness without hiding fairness relevant variation.

## Acknowledgments

We thank the TalentCLEF 2026 organizers for designing and running the shared task and for providing the evaluation infrastructure. We also thank our colleagues who contributed to system development, experiment management, and submission analysis.

## Declaration on Generative AI

During the preparation of this work, the authors used generative AI systems both as components of the submitted retrieval systems and as writing aids. In the submitted systems, large language models were used for text extraction, profile generation, rubric style scoring, and reranking steps. During manuscript preparation, generative AI tools were used for drafting assistance, editing, and language polishing. The authors reviewed and edited the manuscript and take full responsibility for its content [36].

## References

[1] 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.

[2] 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, in: CLEF 2026 Working Notes, 2026.

[3] W. Veys, L. Gasco, M. De Lange, J.-J. Decorte, Overview of TalentCLEF 2026: Task B, Job-Skill Matching with Skill Type Classification, in: CLEF 2026 Working Notes, 2026.

[4] European Commission, What is ESCO?, https://esco.ec.europa.eu/en/about-esco/what-esco, 2026. Accessed 2026-06-01.

[5] R. Nogueira, W. Yang, J. Lin, K. Cho, Document Expansion by Query Prediction, arXiv preprint arXiv:1904.08375 (2019). URL: https://arxiv.org/abs/1904.08375.

[6] L. Gao, X. Ma, J. Lin, J. Callan, Precise Zero-Shot Dense Retrieval without Relevance Labels, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2023, pp. 1762–1777. URL: https://aclanthology.org/2023.acl-long.99/. doi:10.18653/v1/2023.acl-long.99.

[7] 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 and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 3982–3992. URL: https://aclanthology.org/D19-1410/. doi:10.18653/v1/D19-1410.

[8] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, I. Gurevych, BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models, in: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, 2021. URL: https://openreview.net/forum?id=wCu6T5xFjeJ.

[9] N. Muennighoff, N. Tazi, L. Magne, N. Reimers, MTEB: Massive Text Embedding Benchmark, in: Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, 2023, pp. 2014–2037. URL: https://aclanthology.org/2023.eacl-main.148/. doi:10.18653/v1/2023.eacl-main.148.

[10] 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.
[11] J.-J. Decorte, M. De Lange, J. Van Hautte, Multilingual JobBERT for Cross-Lingual Job Title Matching, in: CLEF 2025 Working Notes, volume 4038 of *CEUR Workshop Proceedings*, 2025, pp. 4428–4437. URL: https://ceur-ws.org/Vol-4038/paper_367.pdf.
[12] P. Vachharajani, pjmathematician at TalentCLEF 2025: Enhancing Job Title and Skill Matching with GISTEmbed and LLM-Augmented Data, in: CLEF 2025 Working Notes, volume 4038 of *CEUR Workshop Proceedings*, 2025. URL: https://ceur-ws.org/Vol-4038/paper_375.pdf.
[13] M. Zhang, R. van der Goot, NLPnorth at TalentCLEF 2025: Comparing Discriminative, Contrastive, and Prompt-Based Methods for Job Title and Skill Matching, in: CLEF 2025 Working Notes, volume 4038 of *CEUR Workshop Proceedings*, 2025. URL: https://ceur-ws.org/Vol-4038/paper_377.pdf.
[14] European Commission, Download ESCO Dataset, https://esco.ec.europa.eu/en/use-esco/download, 2026. Accessed 2026-06-01.
[15] X. Zhao, X. Hu, Z. Shan, S. Huang, Y. Zhou, X. Zhang, Z. Sun, Z. Liu, D. Li, X. Wei, Y. Pan, Y. Xiang, M. Zhang, H. Wang, J. Yu, B. Hu, M. Zhang, KaLM-Embedding-V2: Superior Training Techniques and Data Inspire A Versatile Embedding Model, arXiv preprint arXiv:2506.20923 (2025). URL: https://arxiv.org/abs/2506.20923.
[16] Y. Babakhin, R. Osmulski, R. Ak, G. Moreira, M. Xu, B. Schifferer, B. Liu, E. Oldridge, Llama-Embed-Nemotron-8B: A Universal Text Embedding Model for Multilingual and Cross-Lingual Tasks, arXiv preprint arXiv:2511.07025 (2025). URL: https://arxiv.org/abs/2511.07025.
[17] Google AI for Developers, Embeddings in the Gemini API, https://ai.google.dev/gemini-api/docs/embeddings, 2026. Accessed 2026-06-01.
[18] Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, J. Zhou, Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models, arXiv preprint arXiv:2506.05176 (2025). URL: https://arxiv.org/abs/2506.05176.
[19] C. Lee, R. Roy, M. Xu, J. Raiman, M. Shoeybi, B. Catanzaro, W. Ping, NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models, arXiv preprint arXiv:2405.17428 (2024). URL: https://arxiv.org/abs/2405.17428.
[20] A. V. Solatorio, GISTEmbed: Guided In-sample Selection of Training Negatives for Text Embedding Fine-tuning, arXiv preprint arXiv:2402.16829 (2024). URL: https://arxiv.org/abs/2402.16829.
[21] Sentence Transformers, Losses, https://www.sbert.net/docs/package_reference/sentence_transformer/losses.html, 2026. Accessed 2026-06-01.
[22] N. Pipitone, G. Houir Alami, A. Avadhanam, A. Kaminskyi, A. Khoo, zELO: ELO-inspired Training Method for Rerankers and Embedding Models, arXiv preprint arXiv:2509.12541 (2025). URL: https://arxiv.org/abs/2509.12541.
[23] ZeroEntropy, zeroentropy/zerank-2-reranker Model Card, https://huggingface.co/zeroentropy/zerank-2-reranker, 2026. Accessed 2026-06-01.
[24] TechWolf, TechWolf/JobBERT-v3 Model Card, https://huggingface.co/TechWolf/JobBERT-v3, 2025. Accessed 2026-06-01.
[25] Tencent, tencent/KaLM-Embedding-Gemma3-12B-2511 Model Card, https://huggingface.co/tencent/KaLM-Embedding-Gemma3-12B-2511, 2025. Accessed 2026-06-01.
[26] Octen, Octen/Octen-Embedding-8B Model Card, https://huggingface.co/Octen/Octen-Embedding-8B, 2025. Accessed 2026-06-01.
[27] NVIDIA, nvidia/llama-embed-nemotron-8b Model Card, https://huggingface.co/nvidia/llama-embed-nemotron-8b, 2025. Accessed 2026-06-01.
[28] ZeroEntropy, zeroentropy/zembed-1-embedding Model Card, https://huggingface.co/zeroentropy/zembed-1-embedding, 2026. Accessed 2026-06-01.
[29] Microsoft, microsoft/harrier-oss-v1-27b Model Card, https://huggingface.co/microsoft/harrier-oss-v1-27b, 2026. Accessed 2026-06-01.
[30] Qwen Team, Qwen/Qwen3-Embedding-8B Model Card, https://huggingface.co/Qwen/Qwen3-Embedding-8B, 2025. Accessed 2026-06-01.

[31] NVIDIA, nvidia/llama-nv-embed-reasoning-3b Model Card, https://huggingface.co/nvidia/llama-nv-embed-reasoning-3b, 2026. Accessed 2026-06-01.
[32] OpenAI, GPT-5.5 Model, https://developers.openai.com/api/docs/models/gpt-5.5, 2026. Accessed 2026-06-01.
[33] Anthropic, Introducing Claude Opus 4.6, https://www.anthropic.com/news/claude-opus-4-6, 2026. Accessed 2026-06-01.
[34] Google AI for Developers, Gemini API Model Documentation, https://ai.google.dev/gemini-api/docs/models, 2026. Accessed 2026-06-01.
[35] Cohere, Cohere Rerank v4.0 Model Announcement, https://docs.cohere.com/changelog/rerank-v4.0, 2025. Accessed 2026-06-01.
[36] CEUR-WS, Policy on AI-Assisting Tools, https://ceur-ws.org/GenAI/Policy.html, 2025. Accessed 2026-06-01.

# Appendices

These appendices document the main prompts behind the enriched representations and the distilled matching principles, so that the extraction steps can be reproduced at the procedure level. Prompts are reproduced in abridged form; structured outputs are returned as JSON and cached per document.

## A. Skill and task extraction prompt (Task A)

The system prompt (abridged) is shown below. The model returns JSON of the form `{skills:[{name, character, skill_type}], tasks:[{name, description}]}`, which is serialized into the enriched view of Table 1 (role specific skills, transferable skills, and the leading tasks). Each document's view is embedded with a retrieval query or retrieval document task type, and the cosine similarity between a job view and a resume view gives the profile overlap signal used in the internal diagnostic surface (Section 3.1).

```
You are an expert HR analyst specializing in
competency-based talent assessment.

A SKILL is a "can do" capability (a persistent, reusable
competency). Test: "This person NEEDS / HAS X" -> SKILL.
A TASK is a "do" activity (a time-bound action). Test:
"This person DOES / COMPLETED X" -> TASK.
If both tests sound natural, classify as TASK.

Assign each skill exactly one character type from:
  {Trait/Disposition, Thinking/Cognitive, Interpersonal,
   Knowledge/Literacy, Practice/Technique}
and exactly one skill type from:
  {cross_domain, role_specific}.
```

## B. Matching principle distillation prompt (Task A)

For each development query, the gold positives and their embedding nearest hard negatives are assembled into case studies, and a strong model is asked to reverse engineer the labeler's criteria under the abstraction constrained prompt below. The output is a JSON object with a core philosophy, general rules, transferability principles, rejection principles, and seniority and domain policies; representative entries are shown in Table 9. An earlier version conditioned on only five development queries produced title anchored rules (for example, naming specific occupations in accept and reject statements); the abstraction constraint was introduced specifically to remove that anchoring, and the rules are still distilled from the small development set, so we treat them as a transferable prior rather than a tuned mapping.

```
You are reverse-engineering an expert labeler's matching
philosophy. Your output will be applied to job descriptions
you have NOT seen (different industries, titles, domains).
Therefore your rules MUST be abstract and transferable,
not anchored to specific titles or industries.

Express each principle as a CATEGORY of work or a CRITERION,
not as named roles.
  REJECT: "Cashier accepted for Sales Associate role"
  ACCEPT: "Hands-on technical analyst roles transferable to
           similar analytical roles regardless of industry"

Output JSON with: core_philosophy, general_rules,
transferability_principles, rejection_principles,
seniority_policy, domain_policy, decision_framework.
```

## C. Job-title decomposition prompt (Task B)

The query side uses the prompt below; the profile is then serialized as `Job / Tasks / Core skills / Contextual skills`. ESCO concept cards (the corpus side) are built by concatenating each skill's title, description, alternative labels, broader concept category, skill type, and essential and optional occupations.

```
You are an expert in job competency analysis and occupational
skill frameworks (ESCO, O*NET). Given a job title, decompose
it into the skills and competencies required for the role.

Cover five skill types: Knowledge, Practice/Technique,
Cognitive, Interpersonal, Trait/Disposition.

Rules:
  - Skills are NOUNS or noun phrases (capabilities one HAS),
    not actions one DOES.
  - Be specific: "statistical analysis", not "analytics".
  - Return 20-30 skills covering all five types.
  - Mark each skill "core" (essential) or "contextual".

Return JSON:
  {"tasks": [...],
   "skills": [{"name","type","relevance","description"}, ...]}
```

## D. Extracted matching principles (Task A)

**Table 9**
Representative extracted Task A matching principles (abridged from the distilled rule set).

| Category | Principle |
|---|---|
| Core philosophy | Matches are judged on the functional core of daily work activities — the actual tasks, deliverables, and skill clusters exercised — rather than titles, shared keywords, or industry domain. |
| Transferability | Specialist execution roles transfer to broader execution roles in the same functional family when the candidate demonstrates the target's core activities as a subset of their work; doing more does not disqualify doing the specific. |
| Seniority | Seniority is evaluated by actual scope of work rather than title; a candidate may match one level above or below when task descriptions align, but large scope gaps are rejected because the daily work differs. |
| Domain | Industry crossing is freely permitted when the functional skill cluster transfers; domain matching is required only when the role's core function is intrinsically domain specific. |
| Rejection | Shared tools or frameworks alone are insufficient: the primary discriminator is the activity performed with those tools, so a pair is rejected when tools overlap but the work output differs in nature. |