Skillberg at TalentCLEF 2026: Cross-Framework Knowledge-Graph Grounded Multilingual Encoders for Job–Person and Job–Skill Matching

Martin Vielvoye[1], Axel Potier[1,2]

[1]Skillberg, 75 Boulevard Vauban, 59000 Lille, France
[2]ICL, Junia, Université Catholique de Lille, LITL, F-59000 Lille, France

Abstract

We describe Skillberg's submissions to TalentCLEF 2026, covering Task A (Contextualized Job–Person Matching) and Task B (Job–Skill Matching with Skill Type Classification). Both systems are built on a multilingual bi-encoder fine-tuned with contrastive learning over a proprietary multi-framework occupation/skill knowledge graph that spans several major European and international référentiels. Task A combines a bi-encoder trained with a six-stage curriculum, an LLM-based corpus enrichment that augments every job title with a generated description, a hybrid dense/sparse first-stage retrieval, and a language-conditional cross-encoder reranker applied only on the languages where it improves the bi-encoder baseline. Task B combines bi-encoder similarity with a graph-prior rescoring term and a lightweight classifier head that distinguishes Core (essential) from Contextual (optional) skills. On the official final-phase test sets we obtain Avg. MAP(en,es) $= 0.6485$ on Task A and NDCG-graded $= 0.7088$ on Task B, ranking fourth among twenty-seven participating teams on Task A and seventh on Task B. We discuss what worked, what surprised us, and what we would change for TalentCLEF 2027.

Keywords

TalentCLEF, Skill matching, Job matching, Multilingual embeddings, Knowledge graph, Contrastive learning, Reranking

1. Introduction

TalentCLEF is the first community evaluation initiative dedicated to multilingual skill and job-title intelligence for human capital management [1]. The 2026 edition [2] introduces two tasks: Task A on contextualized job–person matching [3], and Task B on job–skill matching with a Core / Contextual skill-type classification step [4]. Both tasks require systems that generalise across languages and across labour-market référentiels — a setting that closely mirrors the production conditions of skills-intelligence platforms deployed across European markets.

About Skillberg. Skillberg is a European AI-infrastructure company building a proprietary multilingual knowledge graph for skills intelligence, headquartered in Lille, France. The platform powers skill- and occupation-matching capabilities for human-resource technology vendors, public-sector employment services, and large enterprises across Europe, with a deliberate emphasis on cross-framework interoperability between major labour-market référentiels and on explainability aligned with the European AI Act. TalentCLEF sits squarely on the technical questions that govern Skillberg's production stack — multilingual semantic matching, robustness across référentiels, and the integration of structured taxonomies with neural retrieval — which is why the team has competed in the benchmark since its 2025 inception. Our 2025 submission to Task A finished first out of fifteen teams; for 2026 we extended the same line of work to cover both tasks of the new edition.

Contributions. The system we describe in this paper rests on three design choices:

CLEF 2026: Conference and Labs of the Evaluation Forum, September 21–24, 2026, Jena, Germany
martin.vielvoye@skillberg.app (M. Vielvoye); axel.potier@skillberg.app (A. Potier)
https://skillberg.app (M. Vielvoye)

1. **Knowledge-graph-grounded curriculum learning, augmented with LLM-generated descriptions.** The bi-encoder is fine-tuned with a six-stage curriculum over training pairs derived from our multi-framework knowledge graph. To compensate for the fact that the Task A corpus exposes only job titles — short and often underspecified — we use a multilingual LLM to generate a description for every title in the corpus, and encode the title-plus-description concatenation.
2. **Hybrid retrieval and language-conditional reranking for Task A.** First-stage retrieval combines bi-encoder cosine similarity with a BM25 lexical signal. A cross-encoder reranker is applied only on the languages where we observed it to improve the bi-encoder baseline in internal evaluation; on the languages where reranking degraded performance we use the first-stage ranking directly.
3. **Graph-prior rescoring and a lightweight type-classifier for Task B.** Bi-encoder similarities are combined with a graph-derived prior over candidate skills; Core vs. Contextual classification uses a lightweight head over the bi-encoder outputs supervised on essential / optional labels from the underlying référentiels.

2. Related Work

TalentCLEF. The TalentCLEF 2025 overview [1] consolidates the methodological observations of the inaugural edition, including the finding that contrastively fine-tuned multilingual encoders consistently outperformed much larger decoder-only LLMs on the title-matching task, and that training strategy mattered more than raw model size. The 2026 edition adds a contextualised job–person formulation [3] and a graded skill-type classification component [4].

Multilingual sentence encoders. Our backbone belongs to the E5 family of contrastively pre-trained multilingual text encoders. The instruction-tuned large variant has shown strong cross-lingual transfer on retrieval benchmarks and serves as the starting point for our fine-tuning.

Skill and occupation matching. A growing body of work treats occupation- and skill-matching as a dense retrieval problem grounded in public taxonomies such as ESCO and O*NET. Recent contributions from the TechWolf and AIDA-UGent groups in particular have made publicly available both fine-tuned encoders and large-scale anonymised career-trajectory data, the latter of which we use under its Creative Commons licence [5].

Cross-encoder reranking. Two-stage retrieval pipelines that pair a bi-encoder retriever with a cross-encoder reranker have become standard for high-precision retrieval. We use a recent multilingual cross-encoder of the Jina v2 family as our reranker.

3. System Overview

Both of our task pipelines share a common bi-encoder backbone, a common knowledge-graph source, and a common contrastive training objective. We describe them once here and then specialise per-task in Sections 4 and 5.

3.1. Backbone encoder

We use `intfloat/multilingual-e5-large-instruct` [6] as the encoder for both tasks. The model is an XLM-RoBERTa-family encoder-only transformer with approximately 560M parameters, pre-trained with a multi-stage contrastive recipe over a large multilingual corpus and instruction-tuned over a mixture of retrieval-style supervised tasks. We follow the model's recommended input convention, prepending the task-type tag (e.g., `query:` / `passage:`) to inputs at both training and inference time.

For Task B we maintain a separate fine-tuned bi-encoder of the same family, dedicated to skill-level inputs; this allows the skill encoder to specialise without compromising the Task A occupation encoder.

3.2. Knowledge graph

The supervision signal for our contrastive fine-tuning comes from a proprietary multi-framework occupation/skill knowledge graph that spans several major European and international référentiels. The graph contains occupation and skill nodes, hierarchical and side relations within each référentiel, and cross-framework alignments connecting equivalent occupations and skills across référentiels. Each node carries multilingual labels covering at least English, French, Spanish, German, Italian, Portuguese, and Chinese. We refer to the cross-framework alignment edges as "junctions" in the remainder of the paper. The exact composition of the graph (number of référentiels, node and edge counts) is not disclosed in this paper.

3.3. Training data

Training pairs are derived from four complementary signals:

- **Intra-référentiel hierarchies.** Broader / narrower relations and sibling structure within each référentiel yield positive pairs of varying granularity.
- **Cross-référentiel junctions.** The alignment edges between référentiels yield cross-framework positive pairs — the same occupation or skill expressed under different taxonomies. These pairs are the principal driver of out-of-distribution generalisation.
- **Multilingual altLabels.** Each canonical label is paired with its localised synonyms across the seven languages above.
- **Skill–occupation co-occurrence.** Skill-pair positives are sampled with oversampling to balance the distribution of frequent against rare skills.
- **Career transitions.** We include positive pairs derived from JobHop [5], an anonymised collection of career trajectories released by AIDA-UGent under CC BY 4.0.

Hard negatives are mined from siblings in the occupation hierarchy and from graph-close neighbours of each positive — nodes that are structurally similar but distinct. We do not disclose the exact pair counts, oversampling ratios, or per-source weights.

3.4. Training objective

Each task is served by its own task-specific bi-encoder, fine-tuned from the shared backbone with a task-appropriate contrastive objective. The Task A encoder uses a combination of GISTEmbedLoss and MatryoshkaLoss within a six-stage curriculum (Section 4.2); the Task B encoder uses CachedMNRL over a unified training set of approximately 850 K pairs (Section 5.1). In both cases the full encoder is updated — we do not use parameter-efficient methods (e.g., LoRA).

3.5. Retrieval index

For inference we encode the candidate corpus once and index the resulting vectors with HNSW [7] approximate nearest-neighbour search. Query vectors are looked up against the index to obtain a top-K candidate list, which is then either returned directly (Task B) or passed to the reranker (Task A; see Section 4.5).

4. Task A — Contextualized Job–Person Matching

4.1. Pipeline overview

Our Task A system has four components: (i) a multilingual bi-encoder fine-tuned with a six-stage curriculum, (ii) an LLM-based corpus enrichment that augments every job title with a generated

description, (iii) hybrid dense–sparse first-stage retrieval that combines bi-encoder cosine similarity with a BM25 lexical signal, and (iv) a language-conditional cross-encoder reranker applied only on the languages where it improved the bi-encoder baseline in internal evaluation.

4.2. Bi-encoder: curriculum learning

The bi-encoder is fine-tuned from `multilingual-e5-large-instruct` (Section 3.1) with a six-stage curriculum. Each successive stage broadens the training data: early stages emphasise straightforward in-framework synonyms; later stages introduce cross-framework alignments, multilingual contrasts, and harder negative-mining regimes. The loss combines GISTEmbedLoss [8] — a guided-in-batch contrastive objective that uses a guide model to select hard negatives within each batch — with MatryoshkaLoss [9], which trains the encoder to produce embeddings that remain informative at multiple truncated dimensions.

4.3. LLM-based corpus enrichment

The Task A corpus exposes only job titles — short, often ambiguous strings on which a bi-encoder has limited surface to align. To give the encoder a richer signal, we use a multilingual LLM (Mistral, accessed through its public API) to generate a one-paragraph description for every job title in the candidate corpus, across the four languages of the broader TalentCLEF setting (English, Spanish, German, Chinese). Each candidate's encoded representation is computed over the concatenation of its title and the generated description, rather than the title alone. We did not generate descriptions for queries at inference time — queries are encoded as provided.

4.4. Hybrid dense–sparse retrieval

First-stage retrieval combines two scores. The dense component is the cosine similarity between the bi-encoder embeddings of the query (a job posting) and each enriched candidate. The sparse component is a BM25 score [10] computed over the title-plus-description text of each candidate. We normalise the two scores on the candidate pool and combine them with a fixed weighting of 0.85 on the dense score and 0.15 on BM25. The dense component does most of the work; BM25 acts as a precision-restorer on rare technical terms, abbreviations, and domain-specific vocabulary that the encoder tends to smooth over.

4.5. Language-conditional reranking

We fine-tuned a cross-encoder of the Jina v2 family on TalentCLEF-style query/candidate pairs, intending to apply it as a second-stage reranker over the top-K of the hybrid retrieval. In our internal evaluation on the official development set, however, the reranker improved precision on German and Spanish queries but consistently *degraded* English and Chinese ones — by enough that applying it unconditionally would have wiped out the gains on the languages where it helped. We therefore apply the reranker only on German and Spanish queries at inference time; English and Chinese queries pass through the first-stage hybrid ranking directly. For the official 2026 Task A test set, this means reranking is applied on Spanish but not on English. We return to this counter-intuitive finding in Section 6.

4.6. Experimental setup

We evaluate on the official TalentCLEF 2026 Task A test sets provided through Codabench, in two languages (English and Spanish). The primary metric is the average of MAP(en–en) and MAP(es–es); we additionally report per-language MAP, MRR, and NDCG.

Table 1
Task A 2026 official final-phase results for Skillberg. Avg. MAP(en,es) is the primary leaderboard metric.

Metric	EN–EN	ES–ES
MAP	0.6597	0.6374
MRR	0.8703	0.8475
NDCG	0.8560	0.8463
Avg. MAP(en,es)	**0.6485**	

4.7. Results

Table 1 summarises our official Task A results on the final-phase test sets.

On the official leaderboard, our submission ranks **fourth out of twenty-seven participating teams** on the primary metric.

5. Task B — Job–Skill Matching with Skill Type Classification

5.1. Skill encoder training

The Task B skill bi-encoder is fine-tuned from `multilingual-e5-large-instruct` (Section 3.1) on a unified training set of approximately 850 K positive pairs aggregated from the knowledge-graph sources described in Section 3.3. The training objective is CachedMNRL [11] (Cached Multiple Negatives Ranking Loss), the gradient-cached variant of in-batch contrastive learning, which lets us train with effective batch sizes larger than would otherwise fit in GPU memory. The encoder is trained end-to-end; we do not use parameter-efficient methods.

5.2. Pipeline

Task B asks systems to retrieve a ranked list of skills from a fixed gazetteer for a given job title, and to label each retrieved skill as either *Core* (essential to the job) or *Contextual* (optional or peripheral). We model the task as a two-step pipeline.

Step 1 — Skill retrieval. We encode the job title context and every gazetteer skill using the Task B bi-encoder of Section 5.1. The final retrieval score for each skill is a linear combination of (i) the cosine similarity between job and skill embeddings and (ii) a graph-prior term derived from the structural relations between the inferred occupation node of the job title and the candidate skill in our knowledge graph. The weight of the graph prior is tuned on a held-out development split.

Step 2 — Core / Contextual classification. For each retrieved skill we apply a lightweight linear classification head, taking as input the concatenation of the job and skill embeddings. The head is supervised by the `essentialSkill` (Core) and `optionalSkill` (Contextual) labels available in the underlying référentiels.

5.3. Experimental setup

We evaluate on the official TalentCLEF 2026 Task B test set provided through Codabench. The primary metric is NDCG-graded; we additionally report NDCG-binary, MAP-graded, and MRR-graded.

5.4. Results

Table 2 summarises our official Task B results on the final-phase test set.

Our submission ranks **seventh** on the primary metric.

Table 2
Task B 2026 official final-phase results for Skillberg.

Metric	Score
NDCG-graded (primary)	**0.7088**
NDCG-binary	0.7487
MAP-graded	0.2736
MRR-graded	0.6451

6. Discussion

What worked: LLM-based corpus enrichment. The single design choice with the clearest impact on Task A was generating a one-paragraph description for every job title in the candidate corpus and encoding the title-plus-description concatenation rather than the title alone. Job titles are short and underspecified by nature; enrichment gives the bi-encoder a much larger surface to disambiguate against, particularly on the long tail of rare or domain-specific titles where surface-form matching is weak. The cost is one offline LLM call per corpus entry — a one-shot expense amortised across all inference queries.

What worked: curriculum learning. A six-stage curriculum — starting from easy in-framework synonym pairs and progressively introducing cross-framework alignments, multilingual contrasts, and harder negative-mining regimes — consistently outperformed single-stage training in our internal evaluations. The gap widened as harder pairs were introduced, suggesting the bi-encoder benefits from progressive exposure rather than uniform sampling.

What surprised us: the cross-encoder reranker hurts in some languages. In two-stage retrieval orthodoxy, a fine-tuned cross-encoder should be a strictly additive precision improvement; reranking narrowly over a top-K from a broad first-stage retriever should not hurt. In our internal evaluation on the official Task A development set, we observed the opposite: while the reranker improved German and Spanish, it consistently *hurt* English and Chinese performance — by enough that applying it unconditionally would have wiped out the gains on the languages where it helped. We therefore submitted a language-conditional reranker (applied on German and Spanish only). We do not have a fully satisfying explanation for this asymmetry; plausible candidates include a mismatch between the cross-encoder's pre-training data distribution and the type of long-tail technical vocabulary present in the corpus, or a residual artefact of our reranker fine-tuning data. We flag this as the most useful per-task lesson for other practitioners building cross-lingual retrieval pipelines: do not assume the reranker is additive across all languages.

What surprised us: hybrid dense–sparse retrieval matters more than expected. Our prior going into Task A was that, with a strong fine-tuned multilingual encoder, BM25 would be redundant — the encoder should already cover the lexical signal as a special case of semantic matching. In practice, even a $0.85/0.15$ weighting in favour of the dense score gave a measurable lift over dense-only retrieval, particularly on queries with rare technical vocabulary, named entities, and corporate-specific abbreviations. Lexical matching is not subsumed by good embeddings; the two signals are complementary.

Per-language behaviour. On the official Task A test set we observed a small but consistent gap between English and Spanish, with English ahead on both MAP (0.6597 vs 0.6374) and NDCG (0.8560 vs 0.8463) — though the gap is narrower on NDCG. We have not yet pinned down whether this reflects label-quality differences in the test set, residual asymmetries in our training-pair sampling, an interaction

with the language-conditional reranker (which is applied on Spanish but not English), or a property of the underlying encoder. A more careful per-language sampling strategy is a clear target for 2027.

Comparison to 2025. Our 2025 system addressed Task A only, used a base-sized encoder, and did not include the LLM-based corpus enrichment, the hybrid retrieval, or the language-conditional reranking in the form used here. The 2026 system extends to both tasks, switches to a large instruction-tuned encoder with a curriculum-staged training schedule, introduces all three of these additions on Task A, and adds the dedicated skill encoder, graph prior, and classifier head for Task B.

Limitations. We did not apply explicit bias mitigation methods to either the training data or the output of either pipeline. TalentCLEF 2026 includes a fairness dimension on Task A; we expect our system to inherit whatever biases are present in the underlying référentiels and in the public encoder's pre-training. A systematic study of these effects is a priority for our 2027 work. We also note that, in keeping with the proprietary status of our knowledge graph, several aspects of the system (graph composition, training-pair counts, hyperparameters) are not disclosed in this paper; this limits independent reproducibility from this manuscript alone, although the public components (backbone encoder, reranker, HNSW index, JobHop data) are sufficient to reproduce qualitatively similar pipelines.

Future work. We see three immediate directions: (i) explicit fairness-aware training and evaluation; (ii) deeper integration of the graph prior into the Task A pipeline (currently it appears only in Task B); (iii) moving from a single bi-encoder per task toward a small family of task-specific encoders sharing a common backbone, in particular a seniority-aware encoder for Task A.

7. Conclusion

We described Skillberg's submission to TalentCLEF 2026 Tasks A and B, built around a shared multilingual bi-encoder fine-tuned with contrastive learning over a proprietary multi-framework occupation/skill knowledge graph. The system ranks fourth out of twenty-seven on Task A and seventh on Task B. The contributions we believe are most transferable to other teams working on similar problems are (i) LLM-based enrichment of short job-title corpora with generated descriptions, (ii) the asymmetric per-language behaviour of cross-encoder reranking and the language-conditional gating that follows from it, and (iii) the persistence of useful lexical signal in dense-retrieval pipelines, even with strong fine-tuned multilingual encoders.

Acknowledgments

We thank the TalentCLEF 2026 organisers, in particular Luis Gascó, Hermenegildo Fabregat, Laura García-Sardiña, and the wider Avature and TechWolf teams who co-organised the lab. This work was partially supported by the Hauts-de-France Region under the agreement signed between the Hauts-de-France Regional Council and the Université Catholique de Lille. We acknowledge AIDA-UGent for releasing JobHop under CC BY 4.0, which contributed to our training data. We also thank the Skillberg engineering team — Nicolas Van-Duysen, Thomas Busin, Tristan Riboulet, Guillaume Brebion, and Merveille Arikungoma — for the production infrastructure that made these submissions possible.

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude (Anthropic) in order to: grammar and spelling check, paraphrase and reword, and improve the writing style. After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

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, in: CLEF (Working Notes), 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, in: CLEF (Working Notes), 2026.

[5] I. Johary, R. Romero, A. C. Mara, T. De Bie, JobHop: A large-scale dataset of career trajectories, arXiv preprint arXiv:2505.07653 (2025). doi:10.48550/arXiv.2505.07653.

[6] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Multilingual E5 text embeddings: A technical report, arXiv preprint arXiv:2402.05672 (2024).

[7] Y. A. Malkov, D. A. Yashunin, Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (2020) 824–836.

[8] A. V. Solatorio, GISTEmbed: Guided in-sample selection of training negatives for text embedding fine-tuning, arXiv preprint arXiv:2402.16829 (2024).

[9] A. Kusupati, G. Bhatt, A. Rege, M. Wallingford, A. Sinha, V. Ramanujan, W. Howard-Snyder, K. Chen, S. Kakade, P. Jain, A. Farhadi, Matryoshka representation learning, in: Advances in Neural Information Processing Systems (NeurIPS), 2022.

[10] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389.

[11] L. Gao, Y. Zhang, J. Han, J. Callan, Scaling deep contrastive learning batch size under memory limited setup, in: Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP), 2021. ArXiv:2101.06983.