QTPride @TalentCLEF 2026: Towards Multi-View Job-Person Matching with LLM Parsing and Knowledge Graph-Enhanced Retrieval

Notebook for the TalentCLEF Lab at CLEF 2026

Huu-Huy-Hoang Tran[1]

[1]University of Engineering and Technology, Vietnam National University

Abstract

This paper describes the QTPride team's approach for TalentCLEF 2026 Task A, focusing on multilingual and cross-lingual job-person matching between job descriptions and candidate resumes. The core challenge lies in the long, noisy, and heterogeneous nature of resumes and job descriptions, where relevant evidence may appear across work experience, education or implicit skill relationships. Our methodology employs a multi-view retrieval framework that first uses an LLM-based preprocessing pipeline to convert unstructured documents into structured JSON representations. We then construct multiple retrieval views, including full text, work experience, ESCO skill graph, explicit skill, and education views, and combine them using reciprocal rank fusion. The combination of full-text, work experience, and ESCO skill graph views achieves the best official test performance among our submitted runs, with average mAP scores of 0.6632 in the monolingual setting and 0.6355 in the cross-lingual setting. Our experiments highlight the potential of combining dense retrieval, structured document parsing, and external labour-market knowledge for job-person matching in the HR domain.

Keywords

Job-person matching, Multilingual retrieval, Knowledge graph, TalentCLEF, Large language models, Human Resources

1. Introduction

Artificial intelligence and natural language processing have become increasingly important in human resource management, especially in digital recruitment [1, 2, 3]. As recruitment platforms generate large amounts of textual data from job descriptions and candidate resumes, automatic job-person matching (CV-JD matching) has become a key task for supporting candidate search and recommendation. Task A of TalentCLEF 2026 [4] addresses this problem by focusing on multilingual and cross-lingual job-person matching across English and Spanish. The objective is to rank candidate resumes according to their relevance to a given job description, leveraging both linguistic and contextual understanding.

CV-JD matching is challenging because both job descriptions and resumes are typically long, noisy, and structurally heterogeneous documents. A job description may contain responsibilities, required qualifications, and preferred skills, while a CV may include work history, education, projects, certifications, and other information that is not always directly relevant to the target job. Moreover, relevant skill evidence is often implicit or expressed through related concepts rather than exact lexical overlap. For example, a job description may mention "machine learning", while a candidate CV may list skills such as "PyTorch" or "TensorFlow". This motivates the integration of structured semantic views and external domain knowledge to capture complementary signals beyond simple lexical overlap.

This paper describes the participation of the QTPride team in TalentCLEF 2026 Task A. We propose a multi-view retrieval framework that combines dense semantic matching and skill knowledge graph information. First, raw job descriptions and CVs are converted into structured JSON representations using an LLM-based preprocessing step. Then, we compute several view-specific matching scores, including full-text similarity, experience similarity, explicit skill similarity, education similarity, and

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

23020073@vnu.edu.vn (H. Tran)

0009-0009-8341-9176 (H. Tran)

a knowledge graph-based skill similarity score. The final ranking is produced by combining these complementary scores. We submitted three runs with increasing levels of structure: a full-text semantic retrieval run, a three-view system incorporating full text, experience, and graph-based signals, and a complete five-view system that further includes explicit skill and education matching views.

2. Methodology

2.1. Preprocessing

The original job descriptions (JDs) and resumes (CVs) are provided as unstructured multilingual text documents with highly diverse formats and writing styles. To obtain more consistent semantic representations, we first convert all documents into structured JSON objects using a large language model (LLM)-based preprocessing pipeline. For each resume, the LLM extracts several semantic fields, including personal information, work experience, education history, technical skills, certifications, and additional information. Similarly, job descriptions are transformed into structured representations containing job information, responsibilities, required skills, preferred skills, and educational requirements.

2.2. Semantic view similarity

We compute four semantic compatibility scores from the structured JSON representations: full-text, work experience, explicit skill, and education. For all four views, the corresponding textual fields are encoded using a sentence embedding model, and cosine similarity between the resulting embeddings is used as the matching score.

Full-text View. The full-text view encodes the complete job description and the complete resume to capture their overall semantic relevance. This view serves as the baseline model.

Work Experience View. The work experience view compares the extracted job responsibilities from the job description with individual work experience entries in the resume. Each experience entry is encoded independently, and the highest similarity score is retained as the final work experience compatibility score.

Explicit Skill view. The explicit skill view measures compatibility using the extracted skill fields from both job descriptions and resumes. The extracted skill lists are encoded into the same embedding space, and their semantic similarity is used as the matching score.

Education view. The education view measures compatibility based on the structured education information extracted during preprocessing. The education fields from both job descriptions and resumes are encoded using the same embedding model to compute the final education compatibility score.

2.3. Graph-based skill similarity

To capture semantic relationships between skills beyond simple textual similarity, we leverage external knowledge from the ESCO taxonomy [5] by constructing a heterogeneous knowledge graph [6]. We build a graph consisting of two types of nodes: skill nodes and occupation nodes. The graph encodes three types of relations: skill-skill relations, hierarchical relations between skills, and occupation-skill requirement relations.

Each node is initialized with a sentence embedding of its ESCO label description. We then apply a Relational Graph Convolutional Network (R-GCN) [7] to learn structure-aware embeddings. The R-GCN is trained through self-supervised learning with a link prediction objective. Specifically, the model is designed to assign higher scores to observed edges in the graph than to randomly corrupted negative edges. Given two node embeddings u and v, the edge score is computed using the dot-product similarity. The model is optimized using a margin ranking loss:

$$\mathcal{L} = \max(0, \gamma - s(u, v^+) + s(u, v^-)) \tag{1}$$

where $s(\cdot, \cdot)$ is the dot-product similarity, and γ is the margin.

After training, each free-text skill extracted from a job description or resume is mapped to the most similar ESCO skill node using cosine similarity between their sentence embeddings. The embeddings of all mapped skill nodes for a document are then mean-pooled to obtain a single graph-enhanced skill representation. Finally, the graph-based compatibility score between a resume and a job description is computed as the cosine similarity between their respective pooled representations.

2.4. Multi-view ranking and Rank fusion

The five views produce complementary rankings that capture different aspects of job-resume compatibility. To combine these signals, we employ Reciprocal Rank Fusion (RRF) [8], a simple yet effective rank aggregation method. Given the ranking position $r_i(d)$ of document d in retrieval view i, the final fusion score is computed as:

$$\mathrm{RRF}(d) = \sum_i \frac{w_i}{k + r_i(d)} \tag{2}$$

where w_i denotes the weight assigned to retrieval view i, and k is a smoothing constant. The fusion weights are selected based on development set performance.

3. Experiments and Results

3.1. Data analysis

3.1.1. TalentCLEF dataset

An official corpus is provided for TalentCLEF 2026 Task A [9] to evaluate job-person matching models. The corpus contains English and Spanish job descriptions (JDs) and candidate resumes (CVs), organized into development and test sets. The development set includes 10 JDs and 472 CVs per language, while the test set contains 40 JDs and 476 CVs per language.

3.1.2. ESCO dataset for skill graph construction

To construct the skill knowledge graph, we use the European Skills, Competences, Qualifications and Occupations (ESCO) dataset as an external source of labour-market knowledge. In this work, we use ESCO concepts as graph nodes and ESCO semantic relations as graph edges. Specifically, skill records and occupation records are used to create skill nodes and occupation nodes, respectively. Relation files are then used to construct edges between skills, between hierarchical skill concepts, and between occupations and their required skills.

Table 1 summarizes the ESCO files used in our graph construction process. The raw ESCO input contains 13,960 skill records and 3,043 occupation records. In addition, we use 5,818 skill–skill relation records, 640 skill hierarchy records, and 126,051 occupation–skill relation records. These relations provide complementary structural information for modeling both explicit and implicit associations among skills and occupations.

Table 1
Statistics of the ESCO data used for skill graph construction.

ESCO component	Records	Attributes	Graph role
Skills	13,960	13	Skill nodes
Occupations	3,043	15	Occupation nodes
Skill–skill relations	5,818	5	Skill–skill edges
Skill hierarchy	640	14	Hierarchical skill edges
Occupation–skill relations	126,051	6	Occupation–skill edges

3.2. Implementation details

For the preprocessing stage, we use GPT-5 mini with a zero-shot prompt to convert multilingual resumes and job descriptions into fixed JSON schemas. The prompt explicitly instructs the model to extract only information explicitly stated in the source documents without inferring or hallucinating missing content. Minor normalization is permitted, such as standardizing date formats, separating comma-separated skill lists, or normalizing obvious entities, while preserving the original semantics and avoiding unnecessary paraphrasing. Outputs are required to conform to the predefined schema, and malformed generations are automatically regenerated. We manually inspected a subset of the extracted outputs to verify that the extracted information remained faithful to the source documents. The complete prompts used for resume and job description parsing are provided in Appendix A.

3.3. Results

3.3.1. Embedding model performance on full text

To determine the baseline embedding model for our retrieval system, we conduct an evaluation on the development set using several multilingual embedding models. The results in Table 2 show that `multilingual-e5-large-instruct` consistently outperforms the other models, achieving the highest mAP scores for both English and Spanish. With an average mAP of 0.8917, it demonstrates the strongest retrieval capability among the evaluated models. Based on this performance, we select `multilingual-e5-large-instruct` [10] as the baseline embedding model for subsequent experiments.

Table 2
Performance of Multilingual Embedding Models on the Development Set (mAP).

Model	English	Spanish	Average
multilingual-e5-large-instruct	**0.8925**	**0.8908**	**0.8917**
multilingual-e5-large	0.8598	0.8807	0.8703
paraphrase-multilingual-mpnet-base-v2 [11]	0.7781	0.7013	0.7397
JobBERT-v3 [12]	0.7281	0.7399	0.7340
paraphrase-multilingual-MiniLM-L12-v2	0.7059	0.6470	0.6765

3.3.2. Monolingual multi-view retrieval performance

After selecting `multilingual-e5-large-instruct` as the strongest embedding backbone on the development set, we further evaluate the contribution of different retrieval views in the proposed multi-view framework. Starting from a full-text semantic retrieval baseline, we incrementally add work experience matching, ESCO graph-enhanced skill matching, explicit skill matching, and education matching. This setting allows us to analyze whether each additional view provides complementary evidence for job-person matching.

We define five retrieval views: full text view (1), work experience view (2), ESCO skill graph view (3), skill view (4), and education view (5). Based on these views, we report five system variants. Run 1 is a baseline that uses only the full text view. The subsequent runs add more views incrementally. Among these variants, Run 1, Run 3, and Run 5 were submitted to the official evaluation system, while Run 2 and Run 4 were used for internal ablation analysis.

As shown in Table 3, adding more retrieval views generally improves performance over the full-text baseline. The work experience view increases the development average from 0.8917 to 0.9030, while adding the ESCO skill graph further improves it to 0.9051 and gives the best official test average among submitted runs, with 0.6632 mAP. Although Run 5 achieves the best development result with 0.9169 mAP, its test average is slightly lower than Run 3, suggesting that the additional skill and education views may introduce useful but less stable signals.

Table 3
Performance of multi-view retrieval runs on the development (mAP) and test (mAP) sets. Submitted runs are marked with *. The best performance is indicated in bold, and the second-best performance is underlined.

Run	Views	Dev			Test		
		En	Es	Avg	En	Es	Avg
1*	(1)	0.8925	0.8908	0.8917	0.6236	0.6274	0.6255
2	(1) + (2)	0.9020	0.9039	0.9030	–	–	–
3*	(1) + (2) + (3)	<u>0.9056</u>	<u>0.9045</u>	<u>0.9051</u>	**0.6622**	<u>0.6641</u>	**0.6632**
4	(1) + (2) + (3) + (4)	0.9055	0.8999	0.9027	–	–	–
5*	(1) + (2) + (3) + (4) + (5)	**0.9155**	**0.9183**	**0.9169**	<u>0.6490</u>	**0.6656**	<u>0.6573</u>

View definitions: (1) full text view; (2) work experience view; (3) ESCO skill graph view; (4) skill view; (5) education view.

3.3.3. Cross-lingual multi-view retrieval performance

TalentCLEF Task A also evaluates the submitted multi-view retrieval variants in the cross-lingual setting. Since only official test results are available for this setting, we report the test mAP scores of the three submitted runs. The same view definitions are used as in the monolingual experiments: (1) full text view, (2) work experience view, (3) ESCO skill graph view, (4) skill view, and (5) education view.

Table 4
Performance of submitted multi-view retrieval runs on the cross-lingual test set.

Run	Views	Test mAP
1*	(1)	0.5894
3*	(1) + (2) + (3)	**0.6355**
5*	(1) + (2) + (3) + (4) + (5)	<u>0.5999</u>

4. Conclusion

In our participation in TalentCLEF 2026 Task A, we explored the effectiveness of a multi-view retrieval framework for multilingual and cross-lingual job-person matching. Our system combines LLM-based document parsing, dense semantic retrieval, and ESCO knowledge graph-enhanced skill matching to capture complementary evidence from resumes and job descriptions. The `multilingual-e5-large-instruct` provided a strong full-text baseline, while the combination of full-text, work experience, and ESCO skill graph views achieved the best official test performance among our submitted runs in both monolingual and cross-lingual settings. Overall, our experiments highlight the potential of combining semantic retrieval with structured job-related views and external labour-market knowledge, while also pointing to the need for more robust cross-lingual alignment and adaptive fusion strategies in future work.

5. Declaration on Generative AI

During the preparation of this work, the author(s) used generative AI tools to assist with language polishing, grammar checking, and improving the clarity of the manuscript. The authors also used large language models as part of the proposed preprocessing pipeline to convert unstructured job descriptions and resumes into structured JSON representations, as described in the methodology section. All experimental design, implementation, result analysis, and final conclusions were conducted and verified by the author(s).

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] W. Veys, L. Gasco, M. De Lange, J.-J. Decorte, Overview of TalentCLEF 2026: Task B — Job-Skill Matching with Skill Type Classification, 2026.

[4] 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.

[5] M. le Vrang, A. Papantoniou, E. Pauwels, P. Fannes, D. Vandensteen, J. De Smedt, ESCO: Boosting Job Matching in Europe with Semantic Interoperability, Computer 47 (2014) 57–64. doi:10.1109/MC.2014.283.

[6] N.-Q. Le, D. D. Hoang, M. V. Tran, T.-H.-Y. Vuong, CareerPathKG: Knowledge Graph Integrated Framework for Career Intelligence, in: Y. Matusevych, G. Eryiğit, N. Aletras (Eds.), Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track), Association for Computational Linguistics, Rabat, Morocco, 2026, pp. 813–822. URL: https://aclanthology.org/2026.eacl-industry.60/. doi:10.18653/v1/2026.eacl-industry.60.

[7] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling Relational Data with Graph Convolutional Networks, in: A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, M. Alam (Eds.), The Semantic Web, Springer International Publishing, Cham, 2018, pp. 593–607.

[8] G. V. Cormack, C. L. A. Clarke, S. Buettcher, 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 '09, Association for Computing Machinery, New York, NY, USA, 2009, p. 758–759. URL: https://doi.org/10.1145/1571941.1572114. doi:10.1145/1571941.1572114.

[9] L. Gasco, H. Fabregat Marcos, C. P. Carrino, L. García-Sardiña, D. Deniz, R. Zbib, J.-J. Decorte, M. De Lange, A. Rodrigo, TalentCLEF 2026 corpus: Skill and Job Title Intelligence for Human Capital Management , 2026. URL: https://doi.org/10.5281/zenodo.18449283. doi:10.5281/zenodo.18449283.

[10] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Multilingual E5 Text Embeddings: A Technical Report, 2024. URL: https://arxiv.org/abs/2402.05672. arXiv:2402.05672.

[11] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, 2019. URL: https://arxiv.org/abs/1908.10084. arXiv:1908.10084.

[12] J.-J. Decorte, M. D. Lange, J. V. Hautte, Multilingual JobBERT for Cross-Lingual Job Title Matching, 2025. URL: https://arxiv.org/abs/2507.21609. arXiv:2507.21609.

A. Prompts for Resume and Job Description Parsing

This appendix provides the exact zero-shot prompts utilized within the LLM preprocessing pipeline to convert unstructured multilingual texts into standardized JSON formats.

A.1. Resume/CV Parsing Prompt

The following prompt template was used to extract structured profiles from candidate resumes:

```
You are an expert resume parser.
Task: Extract structured information from the following resume/CV text and return a
    VALID JSON object only.
Rules:
1. Return ONLY JSON. No markdown, no explanation, no code block.
2. If some information is missing, use null, [] or "" appropriately.
3. Keep the original language/content as much as possible.
4. Do not invent information that is not present in the resume.
5. Normalize obvious entities when possible, but stay faithful to the source text.

Use EXACT structure:
{
  "Personal Information": {
    "name": "",
    "phone": "",
    "email": "",
    "location": ""
  },
  "Professional/Work Experience": [
    {
      "job title": "",
      "company": "",
      "dates": "",
      "responsibilities": []
    }
  ],
  "Education": [
    {
      "degree": "",
      "school": "",
      "graduation date": "",
      "relevant details": ""
    }
  ],
  "Skills": [],
  "Others": {
    "Languages": [],
    "Certifications": [],
    "Additional information": []
  }
}

Resume text:
"""{text_content}"""

Return ONLY valid JSON. No explanation.
```

A.2. Job Description Parsing Prompt

The following prompt template was used to transform unstructured job advertisements into rich hierarchical structures:

```
You are an expert job description parser.
Task: Extract structured information from the following job description text and return
    a VALID JSON object only.
Rules:
1. Return ONLY JSON. No markdown, no explanation, no code block.
2. If some information is missing, use null, [] or "" appropriately.
3. Keep the original language/content as much as possible.
4. Do not invent information that is not present in the job description.
5. Separate responsibilities from requirements carefully.
6. If the JD distinguishes required vs preferred qualifications, preserve that
    distinction.
7. If something is ambiguous and cannot be confidently classified, place it in "Other
    Information" -> "additional_information".
Use EXACT structure:
{
  "Job Information": {
    "job_title": "",
    "company": "",
    "department": "",
    "location": "",
    "employment_type": "",
    "seniority_level": "",
    "salary_range": "",
    "posted_date": "",
    "application_deadline": ""
  },
  "Job Summary": "",
  "Responsibilities": [],
  "Requirements": {
    "required_skills": [],
    "preferred_skills": [],
    "education": [],
    "experience": [],
    "certifications": [],
    "languages": [],
    "domain_knowledge": [],
    "soft_skills": []
  },
  "Benefits": [],
  "Application Information": {
    "application_instructions": [],
    "documents_required": []
  },
  "Other Information": {
    "working_hours": "",
    "remote_policy": "",
    "travel_requirements": "",
    "additional_information": []
  }
}
Job description text:
"""{text_content}"""

Return ONLY valid JSON. No explanation.
```