# MARSAD at TalentCLEF 2026 Task B: Hybrid ESCO-Aware Retrieval for Job–Skill Matching

Shimaa Ibrahim[1], Wajdi Zaghouani[1]

[1]Northwestern University in Qatar, Education City, Doha, Qatar

**Abstract**

This paper describes the **MARSAD** submission to TalentCLEF 2026 Task B, *Job-Skill Matching with Skill Type Classification*. The task requires systems to retrieve and rank the skills associated with a given job title from a predefined ESCO-based skill inventory. Our approach models the problem as hybrid retrieval enriched with taxonomy supervision. Specifically, we combine dense semantic retrieval using `intfloat/e5-base-v2`, lexical matching with BM25, and a training-derived ESCO occupation–skill prior that boosts skills linked to matched occupations. To connect free-form job titles with the structured training taxonomy, we build an ESCO occupation embedding index and apply score boosting through both exact and embedding-based occupation matching. The resulting system ranks the full skill corpus for each query and outputs a TREC-formatted run for evaluation. Our design is lightweight, inference-only, and intended to balance semantic matching, lexical precision, and structured knowledge from the ESCO taxonomy within a single ranking framework. The final submission achieved an official score of 0.6531, improving over the official baseline of 0.6204.

**Keywords**

job–skill matching, ESCO taxonomy, hybrid retrieval, dense retrieval, BM25, TalentCLEF

## 1. Introduction

Advances in Natural Language Processing (NLP) are increasingly being applied to Human Capital Management (HCM), where language technologies can support tasks such as job matching, skill identification, workforce planning, and upskilling strategy design. However, progress in this area depends on the availability of public benchmarks that reflect real labor-market language and allow reproducible evaluation of competing systems. TalentCLEF was introduced to address this gap by establishing a shared evaluation campaign focused on job and skill intelligence, covering benchmark tasks derived from real HCM scenarios [1, 2].

In TalentCLEF 2026 Task B, the objective is to retrieve the skills associated with a given job title and rank them with respect to their relevance to the position. The task is grounded in the ESCO taxonomy, which provides a multilingual structured representation of occupations and skills for the European labour market [3, 4]. Compared with standard semantic retrieval benchmarks, this setting is challenging for at least three reasons. First, job title queries are often short, noisy, and lexically variable. Second, the task requires matching free-form query strings to a controlled skill inventory rather than generating skills freely. Third, the training data provides occupation–skill supervision labelled as *essential* or *optional*, creating an opportunity to exploit structured prior knowledge beyond zero-shot retrieval.

The first edition of TalentCLEF in 2025 showed that Task B was largely approached as a retrieval problem, with participants relying on encoder-based similarity models, lexical matching, contrastive training, re-ranking, and external knowledge sources such as ESCO [1, 2]. These findings motivate retrieval-based systems that can combine semantic similarity with structured occupational knowledge. Our system follows this direction, but instead of fine-tuning a task-specific encoder or training a separate reranker, we propose an inference-only hybrid method that integrates three complementary signals: dense semantic similarity, lexical BM25 scoring, and an ESCO occupation–skill prior derived directly from the training annotations.

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

shimaa.ibrahim@northwestern.edu (S. Ibrahim); wajdi.zaghouani@northwestern.edu (W. Zaghouani)

0009-0000-2044-2306 (S. Ibrahim); 0000-0003-1521-5568 (W. Zaghouani)

More specifically, we represent each candidate skill by merging all of its aliases into a single passage text, encode queries and skills using the E5 bi-encoder, and combine dense similarity with BM25 lexical evidence. We then map each job title to an ESCO occupation through exact title lookup or embedding-based nearest-neighbour matching. When such a match is found, we boost the scores of the corresponding essential and optional skills from the training data, using the `esco_uri` field in the corpus as a direct bridge between training annotations and candidate skills.

Our contribution is therefore a lightweight hybrid retrieval architecture that remains close to the official task formulation while making explicit use of ESCO structure at inference time. The system does not require additional fine-tuning, is efficient enough to run on a single T4 GPU, and produces a full ranked list of candidate skills for each job title query.

The remainder of this paper is organized as follows. Section 2 reviews prior work on TalentCLEF and job–skill matching. Section 3 presents the proposed system in detail. Section 4 describes the experimental setup, Section 5 reports the results, and Section 6 concludes the paper.

## 2. Related Work

Job–skill matching has become an important problem in computational labor-market analysis and HCM, where systems aim to retrieve or rank relevant professional skills for a given job title. TalentCLEF 2025 introduced the first shared benchmark dedicated to job and skill intelligence, covering multilingual job-title matching and job-title-based skill prediction. For Task B, the goal was to retrieve relevant skills for an English job title, and the official evaluation metric was Mean Average Precision (MAP) [1, 2].

The methodological overview of TalentCLEF 2025 shows that retrieval-based methods dominated Task B, often supplemented with prompting, re-ranking, synthetic data generation, or external knowledge such as ESCO [2]. This is consistent with the structure of the task itself, where free-form job titles must be aligned with a controlled skill inventory derived from the ESCO taxonomy [3]. The official benchmark overview also emphasizes that training strategy had a stronger effect than model size alone, highlighting the importance of task-specific adaptation in this setting [1].

Several 2025 Task B submissions explored stronger supervised and hybrid retrieval architectures. Ali [5] proposed LLM-driven data augmentation together with cross-encoder and bi-encoder variants, and reported the strongest published 2025 Task B result among the papers we reviewed, achieving 0.345 MAP on the official test set with a fine-tuned E5-Instruct bi-encoder. Vázquez et al. proposed an ensemble that combines multiple LLM-based representations refined with Sentence-BERT, reporting 0.2442 MAP and explicitly outperforming the official baseline of 0.1874 [6]. Zhang and van der Goot compared discriminative, contrastive, and prompting-based approaches, finding that a fine-tuned classification model performed best for Task B with 0.290 MAP on test and a provisional rank of third out of fourteen teams; they also enriched their models with ESCO titles and descriptions [7].

Our approach differs from these 2025 systems in one important respect. Rather than fine-tuning a new bi-encoder, training a classifier, or adding a separate reranker, we use an inference-only hybrid pipeline that combines an off-the-shelf E5 dense retriever [8], BM25 lexical scoring [9], and an ESCO occupation–skill prior derived directly from the training taxonomy [3]. This design keeps the system lightweight while still exploiting both semantic similarity and structured supervision from ESCO.

## 3. Methodology

### 3.1. System Overview

Our system for TalentCLEF 2026 Task B formulates job–skill matching as a hybrid retrieval problem enriched with ESCO taxonomy supervision. Given a job title query, the system ranks all candidate skills in the provided corpus by combining three signals: (i) dense semantic similarity computed with a pretrained bi-encoder, (ii) lexical similarity computed with BM25, and (iii) a training-derived ESCO occupation–skill prior that promotes skills associated with a matched occupation. The final system

produces a ranked list of all skills in the test corpus for each query and outputs the results in standard TREC format.

### 3.2. Data and Representation

The system uses the official Task B training, validation, and test splits. The training resources include `jobid2terms.json` and `skillid2terms.json` (alias lists for ESCO occupations and ESCO skills, respectively) and `job2skill.tsv` (occupation–skill relevance pairs labelled *essential* or *optional*). In validation mode, the system uses a set of job-title queries, a candidate skill corpus, and a qrels file for local evaluation. In test mode, it uses the official test queries and test corpus.

Each query is represented by the raw job title string. Each skill is represented by its list of aliases from the corpus. Instead of treating aliases as separate candidate documents, we merge all aliases belonging to the same skill into a single text string using a `[SEP]` separator:

$$\texttt{skill\_text} = \texttt{alias}_1\ \texttt{[SEP]}\ \texttt{alias}_2\ \cdots\ \texttt{[SEP]}\ \texttt{alias}_n \tag{1}$$

This produces a one-to-one mapping between corpus skills and passage texts, avoids duplicate candidates caused by alias explosion, and allows the encoder to observe all surface forms of a skill in one representation.

### 3.3. Dense Semantic Retrieval

We use `intfloat/e5-base-v2` as the dense retrieval backbone. Since E5 is trained for asymmetric retrieval, we prepend the query prefix `query:` to job titles and the passage prefix `passage:` to merged skill texts:

$$q = \text{“}\texttt{query: }\text{”} + \texttt{job\_title}, \qquad s = \text{“}\texttt{passage: }\text{”} + \texttt{skill\_text}. \tag{2}$$

All embeddings are generated with `normalize_embeddings=True`, so cosine similarity reduces to a dot product between L2-normalized vectors. Query and skill embeddings are computed on a T4 GPU in large batches (`batch_size = 512`). The dense similarity matrix is then obtained as

$$\text{dense\_sim} = QS^{\top}, \tag{3}$$

where $Q \in \mathbb{R}^{n_q \times d}$ is the matrix of query embeddings and $S \in \mathbb{R}^{n_s \times d}$ is the matrix of skill embeddings.

### 3.4. Lexical BM25 Scoring

To complement semantic retrieval with exact-term matching, we build a BM25Okapi index [10, 9] over the merged raw skill texts. Before indexing, each skill text is lowercased, punctuation is removed, and the remaining tokens are split on whitespace.

For each query, BM25 scores are computed against all candidate skills and min–max normalized to the range $[0, 1]$:

$$\text{bm25\_norm}(q, s) = \frac{\text{BM25}(q, s) - \min_{s'} \text{BM25}(q, s')}{\max_{s'} \text{BM25}(q, s') - \min_{s'} \text{BM25}(q, s')}. \tag{4}$$

The dense and lexical signals are then combined into a hybrid base score:

$$\text{base\_score}(q, s) = 0.7 \cdot \text{dense\_sim}(q, s) + 0.3 \cdot \text{bm25\_norm}(q, s). \tag{5}$$

This score acts as the default ranking signal for all skills and remains the sole score for queries that do not receive an ESCO occupation match.

### 3.5. ESCO Occupation Embedding Index

To connect free-text job title queries with the structured ESCO taxonomy, we build an embedding index over all occupations listed in the ESCO occupation-alias file. Each ESCO occupation is represented by multiple aliases. We encode all occupation aliases with the E5 passage prefix and compute the mean vector for each occupation:

$$\mathbf{v}_{\text{occ}} = \frac{1}{|A_j|} \sum_{a \in A_j} \text{embed}(\text{“passage: ”} + a), \tag{6}$$

where $A_j$ is the alias set of occupation $j$. After averaging, each occupation vector is re-normalized to unit length. We use mean pooling rather than indexing every alias separately because it keeps the occupation index at exactly one vector per ESCO occupation (rather than one per alias), which keeps Path B lookup a single nearest-occupation search instead of a nearest-alias search followed by a de-duplication step; since all aliases of a given occupation denote the same underlying role, this is a natural, low-cost design choice, though we have not separately measured how tightly the alias embeddings of a given occupation cluster.

All occupation aliases are encoded in a single batched GPU encoding call, and the per-occupation means are computed using a vectorized reduction operation. This yields an occupation embedding matrix

$$E \in \mathbb{R}^{n_o \times d}, \tag{7}$$

where $n_o$ is the number of ESCO occupations ($n_o = 3{,}039$ in our training data). Query-to-occupation similarities are then computed by

$$\text{esco\_sim} = QE^{\top}. \tag{8}$$

### 3.6. Training-Derived ESCO Occupation–Skill Prior

The training file `job2skill.tsv` provides ESCO occupation–skill relevance pairs labelled as *essential* or *optional*. From this file, we construct a mapping from each occupation to its associated skills. Instead of using these labels to fine-tune the encoder, we use them as a structured prior at inference time.

We assign a fixed boost value to each label:

$$\text{boost}(s) = \begin{cases} 2.0, & \text{if } s \text{ is labelled essential,} \\ 1.0, & \text{if } s \text{ is labelled optional.} \end{cases} \tag{9}$$

This boost is not used in isolation: when a query is matched to an ESCO occupation (Section 3.7), it is added directly to that skill's dense similarity score to obtain the final ranking score, as defined in Equation (11). For example, a query matched to an occupation with an essential skill whose dense similarity is 0.42 receives a final score of $2.0 + 0.42 = 2.42$; since base_score in Equation (5) is bounded to roughly $[-1, 1]$, this reliably places taxonomy-linked skills above hybrid-only candidates. Skills with no occupation match, or not linked to the matched occupation, keep their unmodified hybrid base_score from Equation (5).

In addition, we construct a lowercase title-to-occupation lookup table from the ESCO occupation-alias file. This allows exact string matching between a query title and a known ESCO occupation alias.

### 3.7. Occupation Matching and Score Boosting

For each query, we attempt to identify a corresponding ESCO occupation in two stages.

**Path A: Exact title match.** The query is lowercased and stripped, then matched directly against the occupation alias lookup table. If an exact match is found, that ESCO occupation is used.

**Path B: Embedding-based occupation match.** If no exact match exists, the system selects the nearest ESCO occupation in embedding space:

$$\hat{j}(q) = \arg\max_{j} \text{esco\_sim}(q, j). \tag{10}$$

This occupation match is accepted only if its cosine similarity is greater than or equal to a threshold $\tau = 0.75$. Queries that do not satisfy this threshold remain hybrid-only and receive no taxonomy-based boost.

When an occupation match is available, the system retrieves the training-linked skills for that occupation and maps them to corpus indices using the `esco_uri` field in the corpus. This provides a direct bridge between the corpus skills and the ESCO skill identifiers used in training.

For each linked skill, the final score is defined as

$$\text{final\_score}(q, s) = \begin{cases} 2.0 + \text{dense\_sim}(q, s), & \text{if esco\_sim}(q, \hat{j}(q)) \geq \tau \text{ and } s \in \text{essential}(\hat{j}(q)), \\ 1.0 + \text{dense\_sim}(q, s), & \text{if esco\_sim}(q, \hat{j}(q)) \geq \tau \text{ and } s \in \text{optional}(\hat{j}(q)), \\ \text{base\_score}(q, s), & \text{otherwise.} \end{cases} \tag{11}$$

This design strongly biases the ranking toward taxonomy-linked skills and generally promotes essential skills above optional and hybrid-only candidates, while still allowing dense similarity to influence the ordering within each group.

### 3.8. Ranking and Submission

After the final scores are computed, all candidate skills are sorted in descending order for each query. In the final submitted configuration, the system ranks and outputs the full skill corpus for every query rather than truncating the list at a small cutoff. This preserves maximum recall and lets the evaluation depend entirely on ranking quality.

The output is written in standard TREC run format:

$$\texttt{query\_id} \quad \texttt{Q0} \quad \texttt{skill\_id} \quad \texttt{rank} \quad \texttt{score} \quad \texttt{run\_tag}. \tag{12}$$

The final run tag used in our submission is `e5_esco_full`. The resulting run file is zipped as a single `.trec` file for Codabench submission.

### 3.9. Validation Procedure

The pipeline includes a validation mode that can evaluate predictions against qrels by constructing a binary relevant set for each query, computing Average Precision per query, and reporting MAP and mean recall, along with the worst-performing queries for diagnostic analysis. However, our team joined TalentCLEF 2026 Task B at the test phase and did not have access to the official validation qrels needed to run this evaluation. As a result, we could not report validation MAP/recall or perform an ablation isolating dense-only, dense+BM25, and dense+BM25+ESCO-prior variants.

Consequently, the occupation-match threshold $\tau = 0.75$ and the dense/lexical fusion weights (Equation (5), 0.7/0.3) were fixed a priori rather than empirically tuned. The threshold was chosen as a conservative cosine-similarity cutoff intended to admit only high-confidence occupation matches and avoid spurious taxonomy boosts. The 0.7/0.3 dense–lexical weighting was chosen to let dense semantic similarity dominate – appropriate for the short, lexically variable job titles discussed in Section 1 – while still letting BM25 contribute exact-term evidence. We report this limitation explicitly to avoid presenting these settings as validation-optimized hyperparameters.

**Table 1**
Summary of the final system configuration.

| Component | Setting |
|---|---|
| Encoder | `intfloat/e5-base-v2` |
| Query format | `query: <job title>` |
| Skill format | `passage: alias_1 [SEP] ... [SEP] alias_n` |
| Dense similarity | Cosine similarity via normalized dot product |
| Lexical scorer | BM25Okapi with min–max normalization |
| Hybrid weights | $0.7\times$ dense $+ 0.3\times$ BM25 |
| ESCO occupation representation | Mean of alias embeddings, then re-normalized |
| ESCO match threshold | $\tau = 0.75$ |
| Essential boost | $+2.0$ over dense similarity |
| Optional boost | $+1.0$ over dense similarity |
| Submission depth | Full corpus ranking |
| Hardware | Google Colab, NVIDIA T4 GPU |
| Run identifier | `e5_esco_full` |

### 3.10. Computational Setup and Carbon Tracking

The system is optimized for execution on Google Colab with an NVIDIA T4 GPU. The main efficiency improvements come from batched GPU encoding of queries, skills, and ESCO occupation aliases, as well as vectorized score computation with NumPy.

We track emissions using CodeCarbon [11] during the model loading and dense encoding stages. The tracker is started before loading the dense encoder and stopped after the query, skill, and ESCO occupation embedding computations are completed. Over this tracked window (12.3 seconds, on a single Tesla T4 GPU with a 2-core Intel Xeon host), the run consumed 0.331 Wh of energy (0.152 Wh GPU, 0.145 Wh CPU, 0.034 Wh RAM) and emitted 0.116 g $CO_2$eq, as reported by CodeCarbon for the tracker's host region. This tracked window covers only model loading and embedding computation, not the subsequent BM25 indexing, scoring, or file-writing steps, so it is a lower bound on the full pipeline's footprint; it is nonetheless consistent with the system's design goal of being lightweight enough to run on a single consumer-grade GPU without additional training. The full emissions summary is saved separately to `emissions.json` and is not included in the Codabench submission archive.

### 3.11. System Configuration Summary

Table 1 summarizes the main system settings.

## 4. Experimental Setup

### 4.1. Task and Data

We evaluate our system on TalentCLEF 2026 Task B, which focuses on job–skill matching with skill type classification. The task is grounded in the ESCO taxonomy and provides official training, validation, and test splits [3, 4]; the training-side files and their contents are described in Section 3. The validation split provides a set of job title queries, a candidate skill corpus, and qrels for local evaluation (Section 3.9). The test split contains 1,139 job title queries and a corpus of 5,358 candidate skills, which form the basis of the final submitted run.

### 4.2. Implementation Details

The same configuration described in Section 3 and summarized in Table 1 was applied at test time, run in Google Colab on a single NVIDIA T4 GPU. The only aspect specific to the final submission is the ranking depth: rather than truncating at a small top-$k$ cutoff, the submitted run ranks the full skill corpus for every test query, preserving maximum recall and leaving ranking quality to determine

**Table 2**
Official competition results.

| System | Official score |
|---|---|
| Official baseline | 0.6204 |
| MARSAD hybrid ESCO-aware system | 0.6531 |

the evaluation score. The resulting output is formatted as a TREC run file and compressed into a ZIP archive for Codabench submission.

### 4.3. Validation Protocol

As described in Section 3.9, we did not have access to the official validation qrels because our team joined the task at the test phase. The final configuration was therefore not selected through validation tuning or ablation; the same a priori configuration described in Section 3 was used to generate the final test submission.

## 5. Results and Discussion

### 5.1. Official Submission Result

Table 2 reports the official competition score for our final system and the baseline run. Our submission achieved a score of 0.6531, improving over the baseline score of 0.6204 by 0.0327 absolute points.

This result shows that the full end-to-end configuration improves over the official baseline, although the single official score does not isolate the individual contribution of dense retrieval, BM25, or the ESCO prior. On the test set, 1,136 of 1,139 queries (99.7%) received an ESCO occupation match and therefore a taxonomy boost – 109 via exact title lookup and 1,027 via embedding-based matching (Path B) – leaving only 3 queries (0.3%) as hybrid-only. The occupation embedding index itself covers 3,039 ESCO occupations built from their alias lists, of which 3,011 have at least one occupation–skill relevance pair in the training data; the `esco_uri` bridge links 5,288 of the 5,358 test corpus skills (98.7%) back to those training annotations.

### 5.2. Discussion

We attribute the improvement over the baseline to three main factors. We frame this as our working hypothesis for what most likely drove the gain, rather than as an established finding: the official result is a single end-to-end score (0.6204 $\rightarrow$ 0.6531), and, as discussed below, we did not run an ablation that isolates the individual contribution of dense retrieval, BM25, or the ESCO prior.

First, the dense E5 retriever provides a stronger semantic representation of both job titles and candidate skills than simpler baseline encoders. Since job title queries are often short and lexically variable, a retrieval-oriented encoder is important for capturing semantic relatedness beyond exact surface overlap [8].

Second, BM25 complements dense retrieval by preserving lexical precision. This is particularly useful for short job titles or technical titles where exact term overlap with skill aliases provides a strong relevance signal [9]. The hybrid score therefore combines the robustness of dense semantics with the precision of term-based retrieval.

Third, the ESCO occupation–skill prior allows the system to exploit structured supervision from the training data without fine-tuning the encoder. By first matching a job title to a likely ESCO occupation and then boosting the associated essential and optional skills, the system can prioritize skills that are already known to be relevant in the training taxonomy. This mechanism is especially useful when queries correspond closely to existing occupational categories in ESCO [3], and on the test set it engaged for 1,136 of 1,139 queries (99.7%; Section 5).

At the same time, the system has some limitations. Most importantly, the occupation-match threshold, the dense/lexical fusion weights, and the essential/optional boost magnitudes were fixed a priori rather than empirically tuned or ablated (Section 3.9), so the single official score above does not by itself isolate which component drives the improvement over the baseline. The local validation evaluator, when used, also treats relevance as binary, which does not fully reflect the distinction between essential and optional skills in the ranking stage. In addition, the ESCO occupation matching step depends on either exact title overlap or a fixed similarity threshold, which may still fail for noisy or highly non-standard job titles. Finally, because the current system is inference-only, it does not benefit from end-to-end fine-tuning or a second-stage neural reranker.

## 6. Conclusion

We presented the MARSAD submission to TalentCLEF 2026 Task B on job–skill matching. Our system formulates the task as hybrid retrieval enriched with ESCO taxonomy supervision, combining E5 dense retrieval, BM25 lexical scoring, and a training-derived occupation–skill prior. The final submission achieved an official score of 0.6531, improving over the baseline score of 0.6204.

The official result indicates that a lightweight inference-time integration of structured occupational knowledge can be competitive and improve over the official baseline without requiring additional fine-tuning, although ablation results are still needed to isolate the contribution of each component.

In future work, we plan to run the pipeline's validation mode on the official validation split (Section 3.9) and use it for a systematic ablation (dense-only vs. dense+BM25 vs. dense+BM25+ESCO prior) that isolates each component's contribution and properly tunes the threshold and fusion weights reported here. We also plan to explore stronger validation metrics that better reflect graded relevance, improved occupation matching for noisy titles, and supervised reranking models that can build on top of the current hybrid retrieval backbone.

## Acknowledgments

This work was funded by the NPRP-C Grant No. 14C-0916-210015 from the Qatar National Research Fund, part of the Qatar Research Development and Innovation Council (QRDI). The findings and conclusions expressed in this paper are solely those of the authors.

## Declaration on Generative AI

During the preparation of this work, the authors used Claude (Anthropic) and ChatGPT (OpenAI) to assist with grammar and style editing, paraphrasing and rewording, and reviewer-response editing. The authors reviewed and edited all content and take full responsibility for the publication's content.

## References

[1] L. Gasco, H. Fabregat, L. García-Sardiña, P. Estrella, D. Deniz, Á. Rodrigo, R. Zbib, Overview of the talentclef 2025: Skill and job title intelligence for human capital management, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction, volume 16089 of *Lecture Notes in Computer Science*, Springer, 2025, pp. 464–485. URL: https://link.springer.com/chapter/10.1007/978-3-032-04354-2_24. doi:10.1007/978-3-032-04354-2_24.

[2] L. Gasco, H. Fabregat, L. García-Sardiña, P. Estrella, D. Deniz, Á. Rodrigo, R. Zbib, Brief overview of talentclef 2025, in: Working Notes of CLEF 2025: Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4038/paper_362.pdf.

[3] European Skills, Competences, Qualifications and Occupations (ESCO) Handbook, European Commission, 2021. URL: https://esco.ec.europa.eu/system/files/2021-07/Handbook.pdf.

[4] L. Gasco, H. Fabregat, L. García-Sardiña, P. Estrella, C. P. Carrino, D. Deniz, A. Rodrigo, R. Zbib, Talentclef at clef2026: Skill and job title intelligence for human capital management, in: Advances in Information Retrieval (ECIR 2026), volume 16486 of *Lecture Notes in Computer Science*, Springer, 2026, pp. 251–258. URL: https://doi.org/10.1007/978-3-032-21321-1_35. doi:10.1007/978-3-032-21321-1_35.

[5] M. Ali, Enhancing job-skill matching with llm-driven data augmentation and fine-tuned bi-encoders, in: Working Notes of CLEF 2025: Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4038/paper_363.pdf.

[6] I. X. Vázquez, R. Sedano, S. González, J. Sedano, Beyond titles: Semantic matching of jobs and skills using llms and s-bert, in: Working Notes of CLEF 2025: Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4038/paper_376.pdf.

[7] M. Zhang, R. van der Goot, Nlpnorth @ talentclef 2025: Comparing discriminative, contrastive, and prompt-based methods for job title and skill matching, in: Working Notes of CLEF 2025: Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4038/paper_377.pdf.

[8] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, F. Wei, Text embeddings by weakly-supervised contrastive pre-training, 2022. URL: https://arxiv.org/abs/2212.03533. arXiv:2212.03533.

[9] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends in Information Retrieval 3 (2009) 333–389. URL: https://www.staff.city.ac.uk/~sbrp622/papers/foundations_bm25_review.pdf. doi:10.1561/1500000019.

[10] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at TREC-3, in: Overview of the Third Text REtrieval Conference (TREC-3), number 500-225 in NIST Special Publication, National Institute of Standards and Technology (NIST), 1994, pp. 109–126. URL: https://trec.nist.gov/pubs/trec3/papers/city.ps.gz.

[11] B. Courty, V. Schmidt, S. Luccioni, Goyal-Kamal, M. Coutarel, B. Feld, J. Lecourt, L. Connell, A. Saboni, et al., mlco2/codecarbon: v2.4.1, 2024. URL: https://zenodo.org/records/11171501. doi:10.5281/zenodo.11171501.