HULAT-UC3M at TalentCLEF: Artificial Intelligence and Natural Language Processing Applied to HR Management

Notebook for the TalentCLEF Lab at CLEF 2025

Alvaro Tejera Villar[1], Isabel Segura Bedmar[2]

[1]Human Language and Accessibility Technologies Group (HULAT), Computer Science and Engineering Department, Universidad Carlos III de Madrid, Av. de la Universidad, 30, 28911 Leganés, Madrid, Spain

Abstract

This paper describes the participation in the TalentCLEF 2025 lab, which focuses on Natural Language Processing methods for Human Capital Management through two tasks: multilingual job title matching (Task A) and job skill prediction (Task B). We explored a range of approaches combining dense semantic representations via sentence embeddings with reranking techniques that leverage Large Language Models (LLMs). In particular, we implemented an LLM-based reranking strategy in which top candidates retrieved via embeddings are reordered based on contextual reasoning. Our systems were designed to tackle two main challenges: the semantic matching of job titles across languages and the accurate prediction of relevant skills in knowledge-intensive scenarios. We also focused on making the models efficient and easy to use in real-world applications. The experiments included both fine-tuned and zero-shot models, tested in several languages and evaluation settings. The work provides insights into the comparative performance of embedding-based and LLM-based methods in multilingual and skill inference scenarios, contributing to a better understanding of their respective strengths and trade-offs in real-world talent management applications.

Keywords

Job Title Matching, Skill Prediction, Sentence Embeddings, Large Language Models, Retrieval-Augmented Generation, Human Capital Management, Multilingual NLP

1. Introduction

The increasing availability of labor market data and the growing demand for personalized talent management solutions have highlighted the importance of Natural Language Processing (NLP) in the field of Human Capital Management (HCM) [1, 2].

In this context, the TalentCLEF 2025 lab [3] proposes two tasks aimed at advancing multilingual information retrieval methods for real-world recruitment and skill identification scenarios. Both tasks are rooted in real-world applications within HCM and use standardized taxonomies such as ISCO [4] and ESCO [5], which are international classification systems for occupations and skills, to ensure relevance and consistency across languages and domains.

The first task (Task A) addresses the challenge of identifying equivalent job titles across different languages. Given a job title as a query, the objective is to retrieve and rank the most similar titles from a predefined list of candidates (the `corpus_elements` file). This list is part of a broader multilingual corpus that includes job titles in English, Spanish, German, and Chinese. The overall corpus is divided into training, development, and test dataset.

The second task (Task B) focuses on predicting the most relevant professional skills for a given job title in English. The system must retrieve and rank the most appropriate skills from a fixed set of candidates also provided in a `corpus_elements` file. This file is derived from a curated corpus of ESCO skills, each enriched with lexical variants and aliases to simulate realistic job description language. As in Task A, the data is organized into training, development, and test datasets. Importantly, while Task A provides training and evaluation data based on multilingual job titles aligned with ESCO and

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*Corresponding author.

100429113@alumnos.uc3m.es (A. T. Villar); isegura@inf.uc3m.es (I. S. Bedmar)

0009-0008-6182-6283 (A. T. Villar); 0000-0002-7810-2360 (I. S. Bedmar)

ISCO taxonomies, Task B follows a hybrid data setup: the training data is derived from structured ESCO sources, whereas the development and test sets are manually annotated. This distinction is crucial, as it introduces realistic and linguistically diverse skill expressions that go beyond formal taxonomy definitions—challenging systems to generalize effectively in practical, knowledge-intensive scenarios.

This paper presents the approaches and results of our participation in both tasks. The core of our methodology explores the combination of dense semantic representations (sentence embeddings) with Large Language Models (LLMs) as rerankers, in a pipeline where top candidates retrieved via embeddings are reordered based on contextual reasoning. While this resembles Retrieval-Augmented Generation (RAG)[6], our system does not perform text generation, but rather uses the LLM to rerank a retrieved list of candidates. This LLM-based reranking approach provides a flexible middle ground between pure semantic retrieval and generative reasoning, enhancing both performance and scalability.

We implemented several model variants for each task and evaluated their performance using the official evaluation metric, Mean Average Precision (MAP), as proposed by the organizers. This allowed us to systematically compare the models and analyze their behavior in multilingual and knowledge-intensive retrieval scenarios.

2. Datasets

The datasets used in TalentCLEF 2025 were specifically designed to reflect real-world scenarios in multilingual job retrieval and skill inference, with data aligned to the ESCO and ISCO taxonomy to ensure consistency across languages and professions.

2.1. Task A – Multilingual Job Title Matching

The dataset for Task A includes job titles in English, Spanish, German, and Chinese. Each title is associated with an ESCO occupation ID and a broader ISCO family ID, enabling semantic grouping across languages. For instance, the job title *air commodore* is linked to the ESCO ID `f2cc5978-e45c-4f28-b859-7f89221b0505` and belongs to the ISCO family `C0110` (Armed Forces Officers).

The scope of the task is restricted to a fixed reference set of 2,500 job titles, provided by the organizers in a file named `corpus_elements`. This file defines the complete retrieval space for the task: each entry consists of a unique identifier and its associated job title. All system predictions must be selected from this predefined list, ensuring consistency and comparability across submissions.

The dataset is divided as follows:

- **Training set:** Contains 15,000 job title pairs per language (English, Spanish, German). No training data is provided for Chinese. Both job titles in the same pair share the same ESCO occupation ID, indicating they are equivalent terms for the same occupation. Each pair is represented by four fields: `family_id` (the corresponding ISCO family), `id` (the ESCO occupation ID), `jobtitle_1` (the text for the first job title), and `jobtitle_2` (the text for the second job title).
- **Development set:** is organized into two files:
  - `queries`: includes a list of 100 job titles. Each entry represents a real-world job title for which the system must retrieve similar titles from the reference corpus (`corpus_elements`).
  - `corpus_elements`: contains the full set of candidate job titles available for retrieval. Each entry in this file includes a unique identifier and the corresponding job title. All predictions must be selected from this predefined set.
  - `q_rels`: Establishes the relationships linking each job title in the `queries` file to its relevant job titles within `corpus_elements`. This file follows the TREC-style format: each line contains a query ID, a candidate job title ID from `corpus_elements`, and a relevance label (1 for relevant, 0 for not relevant).

- **Test set:** Comprises 5,000 job titles per language, with 100 manually selected queries used for evaluation. Unlike the development set, the test set does not include q_rels.

2.2. Task B – Job Title-Based Skill Prediction

The dataset for Task B focuses on predicting skills relevant to English-language job titles, with information drawn directly from the ESCO taxonomy. It is divided into three splits:

- **Training set:** Over 5,000 job titles are linked to their most representative skills. Both job titles and skills are represented by their corresponding ESCO URIs. This set also specifies whether each skill is essential or optional for a given job.
- **Development set:** Comprises 200 job titles, each associated with a curated set of relevant skills. Skills are defined not only by their ESCO ID but also by a list of lexical variants (aliases), simulating realistic and noisy job descriptions.
- **Test set:** Includes 500 job titles for which participants must predict a ranked list of relevant skills.

Although the training set for Task B is constructed using structured information from the ESCO taxonomy, the development and test sets are manually curated. This ensures realistic and diverse skill representations that reflect practical usage scenarios, which is crucial for fair and comprehensive evaluation.

3. Methods

To address the two tasks proposed in TalentCLEF 2025, we designed a flexible architecture combining dense vector representations with LLM-based reranking techniques. Our methodology was guided by the goal of balancing semantic accuracy, multilingual robustness, and computational feasibility. This section presents in detail the two implemented approaches: sentence embedding-based retrieval and LLM-based reranking of retrieved candidates, as well as their shared components such as preprocessing, retrieval, and output handling.

3.1. Method 1: Sentence Embedding-Based Retrieval

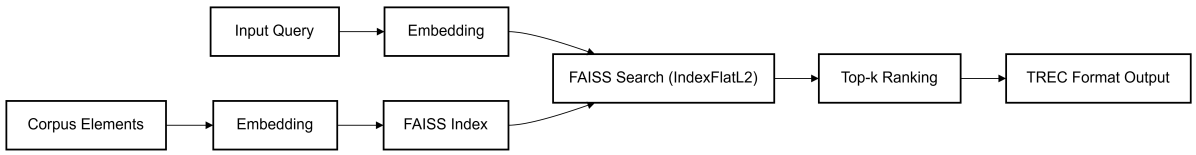

Figure 1: Diagram of the Sentence Embedding-Based Retrieval pipeline. The query and corpus elements are independently embedded into dense vectors using the multilingual model `snowflake-arctic-embed-l-v2.0`, indexed with FAISS, and ranked by semantic similarity.

To semantically represent both queries and the predefined list of candidates (job titles in Task A and skills in Task B), we used a sentence embedding model that maps each textual input into a dense vector within a shared semantic space. This representation enables comparisons based on meaning rather than surface form, making it well suited for semantic retrieval.

We selected `snowflake-arctic-embed-l-v2.0` as our base sentence embedding model. This multilingual model, with 568 million parameters and 1024-dimensional output vectors, was chosen for its top-ranked performance in the retrieval task of the Massive Text Embedding Benchmark (MTEB) [7]. Among all models evaluated, it achieved the highest overall score in this category, making it the strongest available model for retrieval-based applications at the time of writing. The retrieval category in MTEB evaluates how well a model can represent both queries and documents so that items with similar meaning end up close to each other in the model's internal representation. This allows the

system to find the most relevant results based on meaning, not just exact word matches. In our case, this is essential for retrieving job titles or skills that are most closely related to the given query. The model also supports all four target languages of the challenge: English, Spanish, German, and Chinese.

In our pipeline, both queries and the predefined list of candidates were independently normalized—lowercased and stripped of punctuation—and then encoded into dense vectors using the sentence embedding model. These vectors were indexed using FAISS (Facebook AI Similarity Search) [8], a library optimized for fast similarity search in high-dimensional spaces. We used the `IndexFlatL2` structure, which performs exact nearest neighbor search using Euclidean distance. While not optimized for large-scale datasets with millions of entries, it was appropriate given the moderate sizes of the predefined lists of candidates: approximately 2,500 job titles in Task A and 2,500 skills in Task B. To accommodate multilingualism and task specificity, we created five FAISS indices: one per language (English, Spanish, German and Chinese) for Task A, and one for Task B, which was conducted exclusively in English.

We evaluated two embedding configurations. First, a zero-shot setup, where the base semantic embedding model was used without additional training. Second, a fine-tuned version for Task A. The aim of fine-tuning was to adapt the semantic embedding model to better capture subtle semantic differences between job titles.

The training set for Task A contained 15,000 job title pairs per language, where each pair consists of two different job titles that refer to the same ESCO occupation. For example, the titles "pilot officer" and "squadron leader" are distinct surface forms but both map to the ESCO occupation `http://data.europa.eu/esco/occupation/f2cc5978-e45c-4f28-b859-7f89221b0505`. These pairs were used as positive examples to fine-tune the sentence embedding model using a contrastive learning approach.

We employed the `sentence-transformers` library and optimized the model with the `MultipleNegativesRankingLoss` objective. This loss function is particularly effective for retrieval tasks. During training, it takes a batch of positive pairs (e.g., "pilot officer" – "squadron leader", "chief surgeon" – "medical director", etc.) and encourages the model to bring the embeddings of each positive pair closer together in the vector space.

Fine-tuning was performed independently for English, Spanish, and German, resulting in three language-specific sentence embedding models. Each was trained for five epochs using the AdamW optimizer, with a batch size of 32 and a learning rate of 2e-5.

After fine-tuning, these models were used during inference to semantically represent both the input queries and the predefined list of candidates. Each query and each corpus item was passed through the semantic embedding model to obtain a dense vector representation. Then, semantic similarity between the query and all candidate embeddings was computed using cosine similarity. The system ranked the candidate job titles according to their distance to the query in the vector space, assuming that more relevant job titles will have embeddings closer to the query vector. For the Chinese track, no fine-tuning was applied due to the lack of training data, so the base sentence embedding model was used in zero-shot mode.

It is important to understand that the goal of the fine-tuning process is to improve how the model represents the meaning of job titles. The model does not group or classify new queries into categories, nor does it directly say how similar two titles are. What it does is learn to place job titles with similar meanings close together in a shared vector space. Thanks to this, when the system later searches for the most relevant titles, it can simply find those whose vectors are closest to the query—making the retrieval more accurate and better aligned with human intuition.

For Task B, fine-tuning was not applied due to hardware limitations and the large scale of the candidate set. Unlike Task A, the available data for Task B did not consist of pre-defined positive pairs, but rather a large list of skills linked to each job title. Generating meaningful training pairs from this data would have required significant preprocessing and computational resources. As a result, we used the base sentence embedding model (`snowflake-arctic-embed-l-v2.0`) in zero-shot mode. While this limited task-specific adaptation, the model still provided useful semantic representations for retrieving relevant skills via FAISS.

At inference time, semantic similarity between the query vector and corpus vectors was computed using cosine distance. The top-k most similar items ($k = 100$) were selected and ranked. The results were exported in the TREC-style run file format required by the organizers.

3.2. Method 2: LLM-Based Reranking of Retrieved Candidates

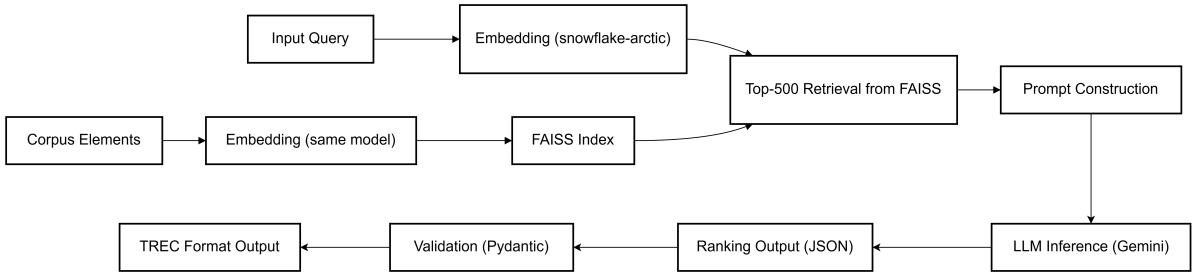

Figure 2: Diagram of the LLM-based reranking pipeline. The top-500 most relevant corpus elements retrieved via FAISS are injected into a prompt and submitted to a Large Language Model (Gemini), which reranks the candidates based on their semantic relevance to the query. No free-text generation is performed.

This method combines embedding-based retrieval with generative ranking using a LLM. Its primary goal is twofold: first, to provide the model with contextual information that it does not possess inherently; and second, to leverage the LLM's inference and reasoning capabilities for ranking semantically relevant items. In this case, the relevant context consists of domain-specific knowledge contained, thereby enhancing performance while minimizing token usage and inference cost. (e.g., job titles or skills), which is not part of the LLM's internal parameters. By retrieving and injecting only the most relevant candidates into the prompt, this approach allows the LLM to operate over a semantically focused subset of external data at runtime, thereby enhancing performance while keeping the input size manageable and reducing inference cost.

As in sentence embedding-based retrieval, the input query was normalized and embedded using the same sentence embedding model. The resulting vector was used to retrieve the top 500 most relevant entries from the FAISS index, built from the corresponding predefined list of candidates. These candidates, along with the original query, were inserted into a structured prompt, which was submitted to the Gemini LLM via API.

Prompts were concise and task-specific. For Task A, the prompt included the original job title and the list of retrieved candidates, with clear instructions to return a ranked list of IDs in valid JSON format. Here is part of our prompt:

```
You are a multilingual job title matching expert. Given the job title: "X",
rank the following job titles in order of most to least similar based on
professional role similarity.

Important Instructions: 1. No gender filtering in reasoning, ignore any
grammatical gender constraints when comparing professional similarity.
2. The query job title may appear in the list, you must exclude it from
your ranking.
3. There might be duplicate job titles with different numbering in the list
below. Treat these duplicates as a single unique job title when ranking.
Only include the first unique occurrence.
4. You (the model) get to decide how many job titles to return. Provide only
the most relevant ones according to professional role similarity, return
at least 10 jobs and a maximum of 110 jobs in the list.
5. Return only valid JSON with the key "ranking", a list of the numbers
(from the provided list) for the job titles you chose in descending order
of similarity (most similar first).
6. You don't need to return 110 jobs, just the most relevant ones according
```

```
to professional role similarity.
7. Don't repeat job titles in the ranking list, just one job title with
different unique IDs.

Return only a valid JSON object in the format:
{"ranking": [list of IDs]}
```

As an additional strategy, we modified the prompt to include the similarity scores computed between the query and each candidate. These scores were provided as guidance for the LLM, helping it refine the ranking, particularly in cases involving near-duplicate or closely related entries.

```
8. Use the provided **similarity score** (lower = more similar) to break
ties or help refine ranking of near-identical job titles.
```

For Task B, we developed an LLM-based reranking system following the same embedding representation and prompting pipeline described for Task A. The prompt was adapted to the skill prediction task: it included the job title and the top 500 retrieved skills from the FAISS index of the predefined list of candidates, along with structured instructions guiding the LLM to rerank and return the most relevant skills in JSON format.

```
You are a multilingual expert in professional skill prediction with deep
understanding of industry job functions.

Your task:
Given the job title: X, select the most relevant skills from the list below
that align with the actual responsibilities, competencies, and expectations
typically associated with the job.

Guidelines:
1. Select no fewer than 10 and no more than 110 skills — choose only those
that are truly necessary for the job.
2. Choose a diverse set of skills reflecting technical, functional, and
soft skills.
3. Ignore surface-level word matches — focus on true professional alignment.
4. Use only the numbers shown before each skill to refer to them.
5. Rank the selected skills from most relevant (first) to least relevant
(last).
6. Do not return duplicate skills.
7. Don't repeat skills in the ranking list; do not include the same skill
with the same c_id more than once. Each skill in skill_list_str has a
unique c_id.
```

Additionally, we tested a full-prompt LLM variant for Task B, in which the complete predefined list of 2,500 candidate skills—including all aliases—was directly included in the prompt. This configuration bypassed the retrieval step entirely, relying on the LLM's ability to reason over the entire candidate set without preselection. We used the same prompt structure as in the LLM-based reranking configuration, but applied it to the full list of candidates. While this approach increased input size and inference cost, it ensured that the model considered all possible skills during ranking. This variant was submitted as an alternative run.

To ensure that the outputs generated by the LLM were structurally valid and compatible with the evaluation scripts, we applied schema-based validation during post-processing. This step enforced a consistent output format—a single JSON object with a `"ranking"` key mapped to a list of unique item IDs—reducing the risk of formatting errors and improving reliability. Using output schemas in LLM pipelines is particularly important when integrating LLMs into structured evaluation workflows, where even minor inconsistencies can lead to parsing failures or incorrect scoring. We enforced a lower bound of 10 elements to guarantee compatibility with `Precision@10` and an upper bound of 110 to support evaluation up to `Precision@100`. These constraints improved robustness and helped limit the length of the LLM's output, reducing inference cost. This method made it possible to work with

just the relevant parts of the data, without having to include the whole dataset in the prompt, which helped reduce cost and improve scalability

4. Resources Employed

The systems developed for both tasks were executed on a local workstation with the following hardware specifications:

- **Processor:** AMD Ryzen 5 3600XT (6 cores, 12 threads)
- **Memory:** 16 GB DDR4 RAM
- **GPU:** NVIDIA GeForce RTX 3060 Ti with 8 GB VRAM

This configuration was sufficient to train fine-tuned sentence embedding models (as done in Task A), generate dense vector indexes with FAISS, and perform inference on moderately sized corpora. In Task B, although no fine-tuning was performed, local embedding and retrieval were still executed efficiently. Most of the experiments were run locally, with GPU acceleration via CUDA used to speed up both embedding and training processes.

In terms of software, we relied primarily on Python, using the following libraries:

- `sentence-transformers` for embedding generation and fine-tuning.
- `FAISS` for dense vector indexing and similarity search.
- `pandas` and `numpy` for data handling and preprocessing.
- `pydantic` for output validation and schema enforcement.

Additionally, we used the Google Gemini API to access the LLM for both our LLM-based reranking and full-prompt systems. Query batching and prompt formatting were handled automatically via a custom wrapper, which also managed error handling and retries.

5. Results and Analysis

This section presents and analyzes the official results obtained by the submitted systems for both Task A (Multilingual Job Title Matching) and Task B (Job Title-Based Skill Prediction), based on the official evaluation metric: *Mean Average Precision (MAP)*. All results correspond to the final test sets held by the organizers, who performed the evaluation and released the MAP scores without disclosing the test data itself.

5.1. Task A – Multilingual Job Title Matching

Task A focuses on semantic matching across languages, requiring systems to handle lexical variation, gendered occupational forms, and cultural nuances. The evaluation was carried out over both monolingual tracks (en-en, es-es, de-de, zh-zh) and cross-lingual tracks (en-es, en-de, en-zh). The submitted models as mention in previous sections are:

- **SE-FineTune-Snow**: Sentence embedding model (`snowflake-arctic-embed-l-v2.0`) fine-tuned for English, Spanish, and German. Chinese was evaluated in zero-shot mode. This approach encodes both queries and candidate job titles into dense vectors and uses cosine similarity to rank the most semantically similar items.
- **SE-ZeroShot**: Same model as above, but used without any task-specific fine-tuning. It also relies on sentence embeddings and cosine similarity to retrieve the most relevant results.
- **LLM-Reranker-1**: A LLM-based reranking based approach that used FAISS to retrieve 500 candidates and passed them to Gemini via a structured prompt.

Table 1
MAP results on the test set for Task A – Multilingual Job Title Matching. Model names abbreviated: SE = Sentence Embedding

Model	Avg. MAP	en-en	es-es	de-de	zh-zh	en-es	en-de	en-zh
SE-FineTune-Snow	0.42	0.479	0.420	0.360	0.433	0.417	0.365	0.394
SE-ZeroShot	0.39	0.478	0.353	0.345	0.433	0.366	0.335	0.394
LLM-Reranker-1	0.18	0.194	0.184	0.173	0.168	0.160	0.178	0.185
LLM-Reranker-2	0.18	0.188	0.188	0.177	0.170	0.183	0.187	0.196

- **LLM-Reranker-2**: Identical to LLM-Reranker-1, but using semantic similarity scores included in the prompt to help the LLM refine its ranking.

Analysis:
As evidenced in Table 1, sentence embedding-based systems consistently outperform LLM-based reranking approaches across all evaluation tracks in Task A. This performance gap is observed in both monolingual (en-en, es-es, de-de, zh-zh) and cross-lingual (en-es, en-de, en-zh) settings, underscoring the relative robustness and reliability of sentence embedding models in multilingual retrieval tasks. The highest MAP score was obtained by the fine-tuned version of the sentence embedding model (`SE-FineTune-Snow`), which highlights the benefits of task-specific adaptation. A detailed analysis of the results yields several relevant insights:

- **Effectiveness of Language-Specific Fine-Tuning:** The marked improvement of the fine-tuned model over its zero-shot variant demonstrates the added value of even modest task-specific training data. Fine-tuning allowed the sentence embedding model to better capture subtle semantic distinctions between job titles in each language, resulting in significantly higher retrieval precision.
- **Robustness and Generalization Across Languages:** Although performance decreases in cross-lingual settings—as expected due to added semantic and syntactic complexity—fine-tuned sentence embedding models maintain relatively strong MAP scores. The strong zero-shot performance on zh-zh (0.433) further illustrates the capacity of the base model to generalize across typologically distant languages, a likely consequence of extensive multilingual pretraining.
- **Limitations of LLM-Based Reranking Approaches:** The comparatively low scores of the LLM-based reranking systems (`LLM-Reranker-1` and `LLM-Reranker-2`), both under 0.19, indicate that LLMs struggle in this retrieval-centered task. This may be attributed to the accumulation of noise in the retrieval stage and to the intrinsic variability of LLM outputs, even when structured prompts and schema validation are employed. Unlike embedding-based approaches, reranking with LLMs relies heavily on the model's interpretation of the input context, which introduces unpredictability and potential drift from the intended ranking criteria.
- **Language-Specific Challenges – The Case of German:** Among the evaluated languages, German consistently yielded the lowest MAP scores. This could be due to its morphological richness, frequent compound word formations, and orthographic variation, which complicate semantic matching. Sentence embedding models may partially mitigate these effects, but the structural properties of the language still pose notable challenges, especially in cross-lingual contexts.
- **Impact of Gendered Language on Evaluation:** A particularly salient issue arises from the use of gendered occupational terms in languages such as Spanish and German. LLM-based systems tend to treat masculine and feminine variants as semantically equivalent, an arguably correct behavior in real-world applications, but this equivalence is not always captured in the gold standard (`q_rels`) relevance annotations.
  To illustrate this misalignment, Table 2 presents a qualitative example based on the development set for Task A, since the official relevance judgments for the test set (`q_rels`) were not publicly

available. Specifically, we use the query "ingeniera de automatización" and compare the top predictions returned by the LLM with a subset of job titles annotated as relevant in the corresponding q_rels file. Since the gold standard provides binary relevance labels without any predefined ranking, we manually extracted the first five relevant job titles as they appear in the q_rels file for this query. This visual comparison reveals that several masculine or lexical variants predicted by the model are not included among the annotated relevant items, despite being semantically appropriate. This suggests that systems focusing on semantic equivalence may be penalized under a strictly lexical evaluation framework.

Table 2
Comparison between LLM predictions and gold standard labels for query "ingeniera de automatización" (q_id 1). The left column lists top-ranked titles predicted by the LLM; the right column shows the job titles annotated as relevant in the gold standard.

LLM Rank	LLM Prediction	Relevant in Gold Standard
1	ingeniero de automatización	ingeniera de sistemas de automatización
2	ingeniera de sistemas de automatización	ingeniera de pruebas
3	ingeniero de sistemas de automatización	coordinador de automatización
4	ingeniera de automatización de procesos	ingeniero de sistemas de automatización
5	ingeniero de automatización de procesos	desarrollador de automatización

Sentence embedding-based models demonstrate strong effectiveness for multilingual job title matching, offering consistent and robust performance across both monolingual and cross-lingual settings. Their reliability, especially when fine-tuned with limited domain-specific data, highlights their scalability and practical utility in real-world retrieval scenarios. By contrast, LLM-based reranking approaches show greater variability and are more sensitive to evaluation design—particularly when semantic equivalence is not explicitly reflected in the relevance annotations.

5.2. Task B – Job Title-Based Skill Prediction

Task B focused on predicting relevant skills from a predefined list of candidates for a given job tittle. Unlike Task A, this task was conducted exclusively in English. We submitted the following approaches:

- **SE-ZeroShot**: Sentence embedding model (snowflake-arctic-embed-l-v2.0) used without task-specific fine-tuning. Both the job title query and candidate skills are encoded into dense vectors, and similarity is computed using cosine distance to rank the most relevant skills.
- **LLM-Reranker**: Combines embedding-based retrieval with LLM-based reranking. The top 500 most relevant skills are retrieved using FAISS, based on cosine similarity with the query embedding, and then re-ranked by the Gemini LLM using a structured prompt that incorporates both the query and retrieved candidates.
- **LLM-FullPrompt**: A non-retrieval approach where the entire list of 2,500 candidate skills is directly embedded into the prompt. The Gemini LLM processes the full input and outputs a ranked list of relevant skills, without relying on prior filtering or similarity scoring.

Table 3
MAP results on the test set for Task B – Skill Prediction.

Model	MAP
SE-ZeroShot	0.112
LLM-Reranker	0.141
LLM-FullPrompt	0.111

Analysis:

As presented in Table 3, the performance differences among the evaluated systems in Task B are less pronounced than those observed in Task A. However, the LLM-Reranker approach achieved the highest MAP score (0.141), while the LLM-FullPrompt model performed slightly worse (0.111), just below the embedding-based system in zero-shot mode (0.112). Despite the narrow margins, these results reveal important contrasts in how embedding-based models and LLM-based approaches address the underlying challenge of predicting relevant skills from job titles—a task that requires not only surface-level similarity but also a degree of contextual and inferential reasoning. The following observations are particularly relevant:

- **Knowledge-Intensive Inference:** Skill prediction is inherently a knowledge-driven task. The relationship between a job title and its associated skills is often implicit and context-dependent, rather than lexically explicit. This places greater demands on models to perform semantic inference, favoring approaches—such as LLMs—that are capable of leveraging contextual cues and domain knowledge beyond what is captured in dense vector representations.
- **Relative Effectiveness of LLM-Based Reranking:** Contrary to its performance in Task A, the LLM-Reranker approach achieves the highest MAP score in this task. This suggests that the combination of targeted retrieval and LLM-driven reranking enables the system to better capture the latent associations between job titles and relevant skills. While the overall improvement is modest, it indicates that this approach can provide meaningful advantages in semantically complex scenarios.
- **Limitations of Sentence Embedding Models in Complex Associations:** The sentence embedding model, used in zero-shot mode, performs similarly to the LLM-FullPrompt model but does not reach the performance of the LLM-Reranker approach. This may be because the model struggles to capture connections between job titles and skills when those relationships are not directly visible in the text. In many cases, knowing which skills go with a job requires background knowledge that goes beyond the words themselves.
- **Context Overload in Full-Prompt LLMs:** The LLM-FullPrompt variant, which processed all 2,500 candidate skills in a single input, did not yield performance gains over the LLM-Reranker configuration. This outcome highlights the practical limitations of large-context inputs: increasing the input size may introduce noise, reduce the model's focus, and exceed effective attention capacity. It underscores the importance of selective context curation when designing prompts for knowledge-intensive tasks. It is also plausible that positional bias played a role: transformer models can tend to allocate more weight to tokens that appear earlier in the prompt, and our skills were fed in a fixed catalogue order that was unrelated to relevance. Although we did not isolate this factor experimentally, such ordering effects could have further blunted the model's effectiveness. Together, these considerations underline the need for selective context curation and careful ordering when constructing prompts for knowledge-intensive tasks.

While no approach achieved particularly high scores in absolute terms, the results suggest that LLM-based reranking approaches hold greater potential for tasks requiring semantic inference over external knowledge sources. Sentence embedding models remain competitive due to their efficiency and simplicity, but may require further adaptation or hybridization to match the performance of context-aware LLM systems. Careful prompt engineering and retrieval design emerge as critical factors for maximizing LLM effectiveness in this setting.

6. Conclusions

The results obtained in both tasks highlight the complementary strengths of sentence embedding-based retrieval and LLM-based reranking approaches in multilingual Human Capital Management scenarios. Sentence embedding models, particularly when fine-tuned, demonstrated robust performance in semantic retrieval tasks such as multilingual job title matching (Task A), combining efficiency,

consistency, and scalability. By contrast, LLM-based reranking approaches showed greater flexibility and capacity for contextual reasoning, particularly in knowledge-driven tasks like skill prediction (Task B). However, these systems are more expensive to run, as they depend on external APIs for LLMs, which often charge based on the amount of text processed. They are also more sensitive to how the prompt is written—small changes in wording or structure can lead to noticeable differences in the results.

Hybrid architectures that integrate sentence embeddings with LLM-based reranking—specifically, our LLM-Reranker approach—emerged as a promising middle ground. This was particularly evident in Task B, the LLM-Reranker system recorded the highest MAP score (0.141), only marginally ahead of both the zero-shot sentence-embedding model and the LLM-FullPrompt configuration. Meanwhile, in Task A, the fine-tuned sentence embedding model achieved the best results (0.42 MAP), demonstrating the benefits of lightweight, language-specific adaptation in multilingual semantic retrieval.

Although LLMs did not perform as well in structured retrieval tasks—like job title matching in Task A—they still offered valuable insights in our experiments. By using LLMs through an API, we were able to explore different prompt formulations, test the impact of including similarity scores, and evaluate alternative ranking strategies, all without needing heavy infrastructure. This flexibility allowed us to experiment quickly and better understand the model's behavior in both well-structured and more ambiguous tasks. However, we also observed that LLM performance was highly sensitive to prompt design and input length, particularly in full-prompt settings like Task B. These observations suggest that while LLMs are powerful, effectively using them for ranking in real-world talent management tasks still requires careful prompt engineering and data curation—more so than is often assumed. Future improvements could involve domain adaptation through lightweight fine-tuning techniques like LoRA or adapter layers, which would allow better performance without large computational costs.

There are several promising directions for future work. One of them is improving the LLM-based reranking approach by building on the strategies already applied. In our current system, we included similarity scores in the prompt to help the LLM rank the most relevant results. This could be further enhanced by adding an intermediate reranking step before prompt generation. A lightweight model trained on a small set of annotated examples could learn to reorder the retrieved candidates based on informative features such as word overlap or semantic similarity. This additional step may help address the relatively low performance observed in Task A, improving the quality of the final candidate list submitted to the LLM.

Another avenue is making the system more adaptive depending on the input. For instance, when the job title is vague or uncommon, the system could rely more on semantic reasoning; in contrast, when the title includes specific or well-known terms, it might prioritize exact word matches. Adjusting this balance dynamically could enhance ranking quality, especially in ambiguous or noisy cases. Finally, Task B could be extended to other languages to evaluate the cross-lingual generalization capabilities of current systems. This would test whether the approach can adapt to different linguistic structures and cultural expressions of professional skills, which is key for real-world multilingual applications.

In addition to architectural improvements, refining evaluation practices could lead to fairer and more realistic assessments. During development—specifically while reviewing the `q_rels` files from the validation set—we observed that some semantically equivalent job titles, particularly gendered variants (e.g., ingeniero vs. ingeniera), were not consistently annotated as relevant. In such cases, LLM-based reranking approaches, which rely on semantic reasoning, may be penalized for returning lexically different but conceptually correct outputs. Although this observation is based on limited evidence and should be interpreted with caution—especially considering the modest performance of LLM-based reranking in Task A—it points to a potential mismatch between system behavior and evaluation criteria. Updating gold relevance annotations to better account for gender variation could improve alignment with real-world expectations, especially in multilingual contexts. Finally, grouping job titles into broader semantic categories could help reduce data sparsity and enhance alignment across languages, particularly in low-resource cases like the Chinese subset, where no training data was available. This would support the development of more robust and scalable multilingual systems for talent management applications.

Another important aspect that deserves further attention is the impact of item ordering in the prompt

on LLM-based ranking. In the reranking runs, we passed the top 500 candidates to the LLM in descending order of cosine similarity, whereas in the Full-Prompt variant every one of the 2500 skills was fed in its fixed catalogue order—neither list was shuffled. Transformer models, however, seem to show some positional bias, often giving more attention to tokens that come earlier in the sequence [9]. Even when the entire prompt fits within the model's context window, this ordering can influence the final ranking output. Recent studies in retrieval-augmented generation and LLM-based QA (e.g., [10]) show that simply reordering retrieved passages can measurably affect generation quality and factual accuracy. This sensitivity exposes a reproducibility and interpretability limitation of current LLM ranking pipelines. Although ordering by cosine similarity provides an inductive bias aligned with retrieval scores, and catalogue order is convenient in the Full-Prompt setup, neither choice appears clearly optimal in light of our results. Future research should therefore test alternative strategies—random shuffling, semantic clustering, or lightweight learned re-ordering—to quantify and mitigate positional effects in both with-retrieval and no-retrieval scenarios. Addressing this limitation is key to building more robust and transparent hybrid architectures for talent-intelligence systems.

In summary, this work highlights the importance of selecting the right combination of semantic retrieval and contextual reasoning depending on the specific demands of the task. While sentence embedding models offer speed, scalability, and solid performance—especially when fine-tuned—LLMs introduce powerful reasoning capabilities that are particularly useful for more abstract or knowledge-intensive tasks. Our findings suggest that hybrid approaches like LLM-based reranking strike a practical balance between these two paradigms. Moving forward, the development of multilingual Human Capital Management systems will benefit from integrating flexible architectures, improving evaluation protocols to reflect real-world semantic variation (e.g., gendered job titles), and enabling better adaptation to diverse linguistic and cultural contexts. These improvements are essential not only to increase model accuracy, but also to ensure fairness, inclusivity, and practical utility in real-world talent intelligence applications across languages and regions.

Acknowledgments

Grant PID2023-148577OB-C21 (Human-Centered AI: User-Driven Adapted Language Models-HUMAN_AI) by MICIU/AEI/ 10.13039/501100011033 and by FEDER/UE.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI's GPT-4 in order to: assist with text translation from Spanish to English, and to improve writing style by suggesting alternative phrasings and enhancing overall clarity and coherence. After using this tool, the authors carefully reviewed, edited, and verified all content to ensure its accuracy and originality, and take full responsibility for the publication's content.

References

[1] E. Hruschka, T. Lake, N. Otani, T. Mitchell, Proceedings of the first workshop on natural language processing for human resources (nlp4hr 2024), in: Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024), 2024.

[2] T. Bogers, D. Graus, M. Kaya, C. Johnson, J.-J. Decorte, T. De Bie, Fourth workshop on recommender systems for human resources (recsys in hr 2024), in: Proceedings of the 18th ACM Conference on Recommender Systems, 2024, pp. 1222–1226.

[3] 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, Springer, 2025.

[4] International Labour Organization, International standard classification of occupations (isco), 2008. URL: https://ilostat.ilo.org/methods/concepts-and-definitions/classification-occupation/, accedido: 2025-05-27.
[5] European Commission, European skills, competences, qualifications and occupations (esco) classification, 2024. URL: https://esco.ec.europa.eu/en/classification, accessed: 2025-05-27.
[6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL: https://arxiv.org/abs/2005.11401. arXiv:2005.11401.
[7] N. Muennighoff, N. Tazi, L. Magne, N. Reimers, Mteb: Massive text embedding benchmark, 2023. URL: https://arxiv.org/abs/2210.07316. arXiv:2210.07316.
[8] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, H. Jégou, The faiss library, 2025. URL: https://arxiv.org/abs/2401.08281. arXiv:2401.08281.
[9] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, P. Liang, Lost in the middle: How language models use long contexts, Transactions of the Association for Computational Linguistics 12 (2024) 157–173. URL: https://aclanthology.org/2024.tacl-1.9/. doi:10.1162/tacl_a_00638.
[10] T. Zhang, D. Li, Q. Chen, C. Wang, L. Huang, H. Xue, X. He, J. Huang, R4: Reinforced retriever-reorder-responder for retrieval-augmented large language models, 2024. URL: https://arxiv.org/abs/2405.02659. arXiv:2405.02659.
[11] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in neural information processing systems 33 (2020) 9459–9474.
[12] L. Gascó, F. M. Hermenegildo, G.-S. Laura, D. C. Daniel, P. Estrella, R. Alvaro, Z. Rabih, Talentclef 2025 corpus: Skill and job title intelligence for human capital management, 2025. URL: https://doi.org/10.5281/zenodo.15292308. doi:10.5281/zenodo.15292308, dataset, version 0.5.0.
[13] D. Deniz, F. Retyk, L. García-Sardiña, H. Fabregat, L. Gasco, R. Zbib, Combined unsupervised and contrastive learning for multilingual job recommendation, in: Proceedings of the 4th Workshop on Recommender Systems for Human Resources (RecSys in HR'24), volume 3788, CEUR Workshop Proceedings, Bari, Italy, 2024. URL: https://ceur-ws.org/Vol-3788/RecSysHR2024-paper_3.pdf.
[14] S. Anand, J.-J. Decorte, N. Lowie, Is it required? ranking the skills required for a job-title, 2022. URL: https://arxiv.org/abs/2212.08553. arXiv:2212.08553.
[15] A. Bhola, K. Halder, A. Prasad, M.-Y. Kan, Retrieving skills from job descriptions: A language model based extreme multi-label classification framework, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 5832–5842. URL: https://aclanthology.org/2020.coling-main.513/. doi:10.18653/v1/2020.coling-main.513.
[16] N. Laosaengpha, T. Tativannarat, A. Rutherford, E. Chuangsuwanich, Mitigating language bias in cross-lingual job retrieval: A recruitment platform perspective, 2025. URL: https://arxiv.org/abs/2502.03220. arXiv:2502.03220.
[17] N. Laosaengpha, T. Tativannarat, C. Piansaddhayanon, A. Rutherford, E. Chuangsuwanich, Learning job title representation from job description aggregation network, 2024. URL: https://arxiv.org/abs/2406.08055. arXiv:2406.08055.
[18] J.-J. Decorte, J. V. Hautte, T. Demeester, C. Develder, Skillmatch: Evaluating self-supervised learning of skill relatedness, 2024. URL: https://arxiv.org/abs/2410.05006. arXiv:2410.05006.
[19] E. Hruschka, T. Lake, N. Otani, T. Mitchell (Eds.), Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024), Association for Computational Linguistics, St. Julian's, Malta, 2024. URL: https://aclanthology.org/volumes/2024.nlp4hr-1/.
[20] L. Gasco, H. Fabregat, L. García-Sardiña, D. Deniz, A. Rodrigo, P. Estrella, R. Zbib, Talentclef at clef2025: Skill and job title intelligence for human capital management, in: C. Hauff, C. Macdonald, D. Jannach, G. Kazai, F. M. Nardini, F. Pinelli, F. Silvestri, N. Tonellotto (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2025, pp. 479–486.
[21] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013. URL: https://arxiv.org/abs/1301.3781. arXiv:1301.3781.

[22] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2014, pp. 1532–1543. URL: https://nlp.stanford.edu/pubs/glove.pdf.

[23] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, 2017. URL: https://arxiv.org/abs/1607.04606. arXiv:1607.04606.

[24] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised learning of universal sentence representations from natural language inference data, 2018. URL: https://arxiv.org/abs/1705.02364. arXiv:1705.02364.

[25] D. Cer, Y. Yang, S. yi Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, R. Kurzweil, Universal sentence encoder, 2018. URL: https://arxiv.org/abs/1803.11175. arXiv:1803.11175.

[26] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL: https://arxiv.org/abs/1908.10084. arXiv:1908.10084.

[27] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training, OpenAI Blog 1 (2018). URL: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[28] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: https://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[29] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL: https://arxiv.org/abs/1910.10683. arXiv:1910.10683.

[30] K. Enevoldsen, I. Chung, I. Kerboua, M. Kardos, A. Mathur, D. Stap, J. Gala, W. Siblini, D. Krzemiński, G. I. Winata, S. Sturua, S. Utpala, M. Ciancone, M. Schaeffer, G. Sequeira, D. Misra, S. Dhakal, J. Rystrøm, R. Solomatin, Ömer Çağatan, A. Kundu, M. Bernstorff, S. Xiao, A. Sukhlecha, B. Pahwa, R. Poświata, K. K. GV, S. Ashraf, D. Auras, B. Plüster, J. P. Harries, L. Magne, I. Mohr, M. Hendriksen, D. Zhu, H. Gisserot-Boukhlef, T. Aarsen, J. Kostkan, K. Wojtasik, T. Lee, M. Šuppa, C. Zhang, R. Rocca, M. Hamdy, A. Michail, J. Yang, M. Faysse, A. Vatolin, N. Thakur, M. Dey, D. Vasani, P. Chitale, S. Tedeschi, N. Tai, A. Snegirev, M. Günther, M. Xia, W. Shi, X. H. Lù, J. Clive, G. Krishnakumar, A. Maksimova, S. Wehrli, M. Tikhonova, H. Panchal, A. Abramov, M. Ostendorff, Z. Liu, S. Clematide, L. J. Miranda, A. Fenogenova, G. Song, R. B. Safi, W.-D. Li, A. Borghini, F. Cassano, H. Su, J. Lin, H. Yen, L. Hansen, S. Hooker, C. Xiao, V. Adlakha, O. Weller, S. Reddy, N. Muennighoff, Mmteb: Massive multilingual text embedding benchmark, arXiv preprint arXiv:2502.13595 (2025). URL: https://arxiv.org/abs/2502.13595. doi:10.48550/arXiv.2502.13595.

[31] S. Colvin, the Pydantic Team, Pydantic: Data validation and settings management using python type annotations, https://docs.pydantic.dev/, 2025. Accessed: 2025-05-27.