COTECMAR–UTB at TalentCLEF 2025: Linking Job Titles and ESCO Skills with Sentence Transformer Embeddings

Notebook for the TalentCLEF Lab at CLEF 2025

Jhonattan Llamas[1,2,*], Edwin Puertas[1], Jairo Serrano[1] and Juan Martinez[1]

[1]*Universidad Tecnológica de Bolívar, Faculty of Engineering, Cartagena de Indias 130001, Colombia*
[2]*Corporación de Ciencia y Tecnología para el Desarrollo de la Industria Naval, Marítima y Fluvial (COTECMAR), Bolívar, Cartagena D.T. y C., Colombia*

Abstract

This paper describes the COTECMAR–UTB submission to Talent-CLEF 2025TaskB, which focuses on retrieving the skills most relevant to a given job title. Our approach is a lightweight, unsupervised pipeline that normalizes job titles and skill aliases, encodes both with the Sentence-Transformer `paraphrase-multilingual-mpnet-base-v2` into a shared 768-dimensional space, and ranks candidate skills by cosine similarity. The model is used strictly in a zero shot with no fine tuning so the system is easy to reproduce and deploy. On the official Codabench leaderboard our run ranked 19th of 26 participants, outperforming the organizer's baseline by two positions. Although the gap with top systems remains considerable, the result confirms that a multilingual off the shelf encoder, combined with simple text enrichment through synonym lists, can deliver competitive performance without task specific training. We analyze strengths and failure cases of our system and compare it against recent state of the art methods that leverage supervised fine-tuning, contrastive learning, or graph-based modeling. Our findings highlight practical direction, such as skill frequency reweighting and knowledge graph integration for boosting retrieval accuracy while preserving the portability of zero shot embeddings.

Keywords

Sentence transformers, Multi-label classification, Natural language processing, ESCO dataset, Human resources, Job recommendation, Job matching, Job skills

1. Introduction

The management of Human Resources (HR) is essential to every organization. A human resource is any person who is compensated for providing skills or knowledge that help an organization achieve its business goals [1]. The HR department assumes this responsibility by overseeing workforce development and performance, strengthening employees' skills and capabilities, and ensuring compliance with International Labour Organization (ILO) standards [2]. In doing so, HR enhances the organization's competitiveness and overall effectiveness and manages the entire recruitment cycle, from posting job offers to interviewing and selecting the most qualified candidates [3]. The process of posting job openings and selecting candidates is critical for hiring the right person for the role the organization needs. Successful hiring, however, depends on a clear understanding of the current labour market, a precise definition of each position, and an explicit identification of the skills the role demands.

To streamline this matching process across languages and data sources, a standardized taxonomy of occupations and skills is indispensable. The European Skills, Competences, Qualifications and Occupations (ESCO) taxonomy offers a multilingual, hierarchical classification of thousands of occupations and skills, maintained by the European Commission [4]. By normalizing both job titles and skill labels to ESCO, HR systems can guarantee consistency and interoperability.

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain
*Corresponding author.
llamasj@utb.edu.co (J. Llamas); epuerta@utb.edu.co (E. Puertas); jserrano@utb.edu.co (J. Serrano); jcmartinezs@utb.edu.co (J. Martinez)
https://github.com/EdwinPuertas (E. Puertas)
0009-0005-9993-5061 (J. Llamas); 0000-0002-0758-1851 (E. Puertas); 0000-0001-8165-7343 (J. Serrano); 0000-0003-2755-0718 (J. Martinez)

At the same time, socio-technological landscapes are evolving at an unprecedented pace, industries and workplaces are rapidly changing. Technological advances, such as task automation and Artificial Intelligence (AI), are reshaping the labor market by creating new roles that demand specialized skills, often difficult to source [5]. Recent progress in Natural Language Processing (NLP), particularly the Transformer architecture [6], has shown that language models can achieve reliable accuracy in mapping job titles to skills. Building on this insight, we adopt the multilingual sentence-transformer *paraphrase-multilingual-mpnet-base-v2*, which produces high-quality embeddings while remaining computationally efficient. By combining these embeddings with synonym enrichment from ESCO in an embedding-based retrieval pipeline, our system delivers accurate skill predictions for a given job title with minimal computational overhead.

2. Related works

NLP has become integral to contemporary HR analytics, powering applications that range from automated parsing of curriculum vitae to candidate–job matching. Résumé analyzers, for instance, extract structured information—education, experience, and skills—from otherwise unstructured CVs, allowing recruiters to screen applicants efficiently. Furthermore, semantic matching techniques compare candidate profiles with job descriptions, improving recommendation accuracy and fit scoring [7].

2.1. Pre-trained Transformers in Job Title Embeddings

The rise of transformer-based language models has significantly influenced job-title embedding and job–skill matching. The Transformer architecture enabled context-aware representations that outperform earlier static embeddings, laying the groundwork for modern approaches [6]. Building on BERT-style encoders, early HR-domain work began to model job titles with pre-trained transformers. For example, JobBERT performed a generic BERT model with information about co-occurring skills to better capture job title semantics [8]. By incorporating skill labels extracted from job postings, JobBERT achieved considerable improvements in job title normalization tasks over using a generic sentence encoder. This demonstrated that skills can enhance transformer-based title embeddings, opening the door for more specialized models in recruitment AI.

2.2. Skill-Augmented and Unsupervised Methods (2021–2023)

Subsequent methods leveraged skill data more directly to learn job title representations without extensive manual labeling. Zbib et al. (2022) proposed an unsupervised learning approach using "noisy" skill labels for training a job title similarity model [9]. Instead of curated title pairs, they used skill co-occurrence in job postings as a weak signal for title similarity. This approach proved as effective as supervised techniques while avoiding costly labeled data, yielding strong results on job title ranking and normalization tasks. In parallel, Anand et al. (2022) tackled the job–skill matching from the opposite angle: given a job title, predict its required skills [10]. Their model used a pre-trained Language-agnostic BERT Sentence Encoder (LaBSE) to rank skills by importance, under the finding that important skills appear frequently across similar job titles. By training on large-scale title–skill associations with weak supervision, the model learned to predict skill importance for a job title and even generalized well to other languages [10]. These works confirmed that skill information – whether used to learn title embeddings or to infer relevant skills – can substantially improve matching performance.

Further extending skill-based representation learning, VacancySBERT was introduced, a Siamese network to jointly embed job titles and skills [11]. They trained a RoBERTa-based model on an enormous dataset of title–description pairs with a novel objective linking titles to the skills mentioned in their job descriptions. The resulting encoder outperformed both generic encoders (e.g. BERT, Sentence-BERT) and prior job-specific models. Notably, VacancySBERT achieved a 10% improvement (and up to 21.5% when skill features are included) in recall@K over baseline methods. This marked a significant

leap in distantly-supervised job title embeddings, showing that massive-scale training on title–skill co-occurrences can yield very robust representations.

2.3. Multilingual and Contrastive Learning Advances (2022–2024)

As the field matured, focus expanded to multilingual and more sophisticated training regimes. Anand et al.'s use of LaBSE hinted at multilingual capability [10], but a dedicated solution arrived with Deniz et al. (2024)'s multilingual job title encoder [12]. They introduced a two-stage learning strategy: first an unsupervised pre-training on skill and title co-occurrence across 11 languages, followed by a supervised contrastive fine-tuning using known similar/dissimilar title pairs from the ESCO skill/job taxonomy [9]. This combined strategy yielded a state-of-the-art multilingual job title representation model. In fact, the new model achieved a 4.3% improvement in mean Average Precision (MAP) on English rankings over the previous best monolingual model, and it maintained strong performance in all 11 tested languages [12]. The learned embedding space also exhibited excellent cross-lingual alignment, enabling job matching across languages. This was a crucial step as it globally aligned job title semantics, allowing, for example, a French job title to be matched to its English equivalent via vector similarity.

Meanwhile, other work explored alternative sources of supervision beyond skill tags. Laosaengpha et al. (2024) proposed learning job title representations directly from their full job descriptions instead of only the extracted skills [13]. They designed a Job Description Aggregation Network that encodes the rich free-text content of job descriptions and uses a bidirectional contrastive loss to align each title with its description in embedding space. This approach captures additional context (e.g., responsibilities and qualifications) beyond just skill keywords. It outperformed a purely skill-based title encoder on both in-domain and cross-domain evaluations, demonstrating that aggregating detailed job description text can yield more informative title embeddings [13]. Together, these advances in contrastive training and multilingual modeling have pushed job matching systems to be more accurate and applicable across global job markets.

2.4. Knowledge Graphs and Cross-Domain Enhancements (2024–2025)

The most recent generation of solutions integrates structured knowledge and extends beyond job titles alone. Fabregat et al. (2024) introduced an inductive graph neural network (GNN) that combines textual embeddings with a job–skill knowledge graph [14]. In their framework, job titles and skills are nodes in a graph (e.g., derived from a taxonomy like ESCO), and a GNN propagates information between related jobs and skills. This inductive GNN approach enables the model to learn from both unstructured text (job title descriptions) and structured relationships (skill taxonomy). As a result, it can make more accurate job and skill recommendations, even handling previously unseen skills or titles by leveraging the graph context [14]. This graph-enhanced method highlights the value of domain ontologies in improving AI-driven matching.

In a similar vein, Alonso et al. (2025) explicitly leveraged an expert curated knowledge base – the U.S. O*NET database – to improve job matching and skill recommendation [15]. O*NET provides a structured mapping of occupations to required skills. The proposed approach uses transformer-based text representations of job postings in conjunction with the O*NET occupational data to recommend relevant skills and match candidates. By anchoring textual analysis to a knowledge base, the system benefits from authoritative job-skill relationships. Alonso et al. report that this knowledge-infused transformer model yields better alignment between job descriptions and required skills, and aids in matching job titles to suitable candidates [15]. This represents a trend of combining AI with human-curated knowledge for superior results.

Finally, researchers have begun to bridge job matching across document types, not just title-to-title comparisons. Rosenberger et al. (2025) presented CareerBERT, a model that maps both resumes and job titles into a unified embedding space for recommendation tasks [16]. CareerBERT is built on transformer encoders fine-tuned to align a candidate's resume with relevant ESCO job titles, enabling personalized job recommendations. Through a two step evaluation (automated and human expert

review), CareerBERT was shown to outperform prior embedding-based approaches to job matching, while providing robust, meaningful recommendations in practical settings. In particular, it significantly outscored traditional methods when matching resumes to open positions, and human recruiters found its suggestions more relevant on average [16]. This exemplifies the cutting edge of the field: using transformer models to integrate rich textual data (like resumes) with standardized job representations, thereby extending the scope from job–job similarity to candidate–job matching.

In summary, the field of job title embedding and job skill matching has evolved from early text similarity methods to sophisticated AI models leveraging transformers. Progress over the past few years is marked by incorporating domain specific information (skills, descriptions), adopting unsupervised and contrastive learning to overcome data scarcity, expanding to multilingual contexts, and integrating knowledge graphs and databases for context. The latest approaches even cross traditional boundaries to align different data modalities (resumes vs. job postings). This chronological progression underscores a clear trend: increasingly specialized and hybrid AI techniques are pushing the state of the art in linking job titles with the skills and candidates that define them.

3. Methodology

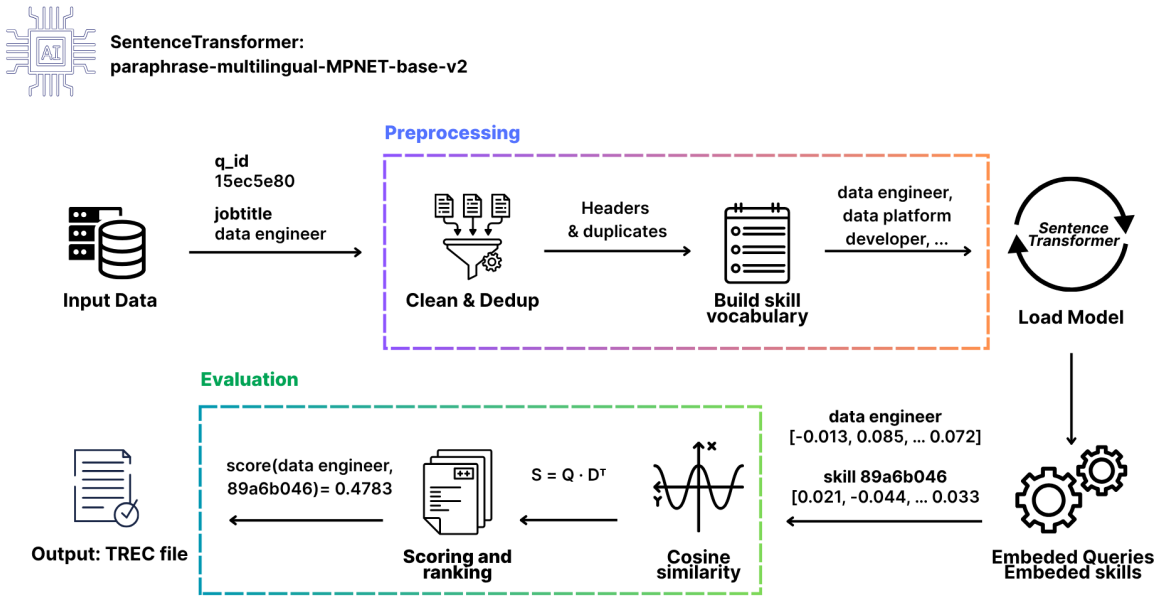

Figure 1: Diagram of the System's Pipeline

Our system is an embedding-based pipeline (Figure 1) that retrieves the most relevant skills for a given job title by encoding both titles and skills into a shared 768-dimensional semantic space. We employ the pretrained Sentence-Transformer `paraphrase-multilingual-mpnet-base-v2` as our backbone because, in our internal validation, it yielded the highest Mean Average Precision (MAP) among all models we tested, while public benchmarks also report strong average performance across sentence-embedding and semantic-search tasks. The model is used strictly in a zero-shot without additional fine tuning performed on the TalentCLEF corpus, so its role is purely inferential. Table 1 summarizes benchmark scores, inference speeds, and model sizes of leading pre-trained transformers; `paraphrase-multilingual-mpnet-base-v2` combines competitive speed with the best empirical MAP in our experiments. The subsections that follow describe each stage of the pipeline in detail, from input normalization to the cosine-similarity ranking that produces the final ordered list of predicted skills.

Table 1
Pre-trained Sentence-Transformer Models: Benchmark Results and Resource Footprint

Model Name	Avg. Perf.	Speed (sent/s)	Size (MB)
all-mpnet-base-v2	63.30	2 800	420
multi-qa-mpnet-base-dot-v1	62.18	2 800	420
all-distilroberta-v1	59.84	4 000	290
all-MiniLM-L12-v2	59.76	7 500	120
multi-qa-distilbert-cos-v1	59.41	4 000	250
all-MiniLM-L6-v2	58.80	14 200	80
multi-qa-MiniLM-L6-cos-v1	58.08	14 200	80
paraphrase-multilingual-mpnet-base-v2	**53.75**	**2 500**	**970**
paraphrase-albert-small-v2	52.25	5 000	43
paraphrase-multilingual-MiniLM-L12-v2	51.72	7 500	420
paraphrase-MiniLM-L3-v2	50.74	19 000	61
distiluse-base-multilingual-cased-v1	45.59	4 000	480
distiluse-base-multilingual-cased-v2	43.77	4 000	480

3.1. Input Data

The pipeline begins by ingesting the provided Task B dataset files. We used the JSON mappings of job and skill identifiers to their textual variants: jobid2terms.json and skillid2terms.json. The jobid2terms.json file provides a set of lexical variants (synonyms and alternative phrasings) for each job title identifier. For example, the ESCO occupation ID corresponds to variants like “application developer”, “application programmer”, “software developer”, “software engineer”, etc. Similarly, the skillid2terms.json file lists textual aliases for each skill identifier (ESCO skill concept).

For instance, a given skill ID might include variants such as “web services” and “web services systems” as synonymous terms for that skill. In addition to these vocabulary resources, the input also includes the list of query job titles (from the validation or test set queries file) for which the system must predict relevant skills, and the full set of candidate skills. This stage outputs the raw textual data: lists of job title terms and skill terms, indexed by their identifiers.

3.2. Cleaning and Deduplicate

Before encoding the texts, we perform normalization and deduplication on the collected terms. All job title and skill terms are lowercased and stripped of extraneous punctuation and spacing to ensure consistency. We also remove any duplicate entries or trivial variants within the lists of terms. For example, if a skill’s alias list contained both a singular and plural form of the same phrase, or exact duplicated phrases, these would be consolidated to a single representative term.

This cleaning step yields a refined set of unique job title phrases and skill phrases. The cleaned data maintains mapping to the original IDs, ensuring that each job and skill identifier is associated with a clear set of distinct textual representations. The output of this stage is a cleaned dictionary of job title variants and a cleaned dictionary of skill variants, ready for embedding.

3.3. Build Skill Vocabulary

In this stage, we construct the skill vocabulary that will be used for retrieval. Using the cleaned skillid2terms data, we aggregate all unique skill alias phrases and organize them by skill ID. Each skill in the vocabulary is thus represented by a set of one or more aliases – e.g., a skill ID might be associated with the terms “pricing strategies”, “pricing tactics”, “pricing plans”, etc., as given in the dataset.

The output of this stage is a complete list of skill entries (each with an ID and list of alias terms) that will be fed into the embedding model. This forms the “corpus” of documents (skills) to be retrieved. We note that a similar vocabulary is inherently defined for job titles via jobid2terms, but since in our task

the queries are single job titles provided in the test set, the job vocabulary primarily serves to supply possible synonyms for better query representation.

3.4. Load Model

We then load the Sentence-Transformer model paraphrase-multilingual-mpnet-base-v2 into memory. This model maps sentences or phrases to a 768-dimensional dense vector space (embedding). It has been pre-trained on large-scale paraphrase data and indicates great performance across various languages, helping us to create semantic relations between words, as shown in Table 1. We do not perform any further training or fine-tuning on this model; it is used directly to encode texts in an off-the-shelf manner. Loading the model yields a ready-to-use encoder capable of transforming any input phrase (job title or skill phrase) into its vector representation. This stage's output is the loaded model instance, which will be used in the subsequent embedding stages.

3.5. Embed Queries (Job Titles)

In the query embedding stage, each input job title query is converted into an embedding. For a given job title query, we first identify its canonical ID or find its synonyms in the jobid2terms mapping (if available). If the query exactly matches or is contained in the list of variants for a job ID, we leverage all those lexical variants to enrich the query representation. Specifically, we encode each variant phrase of the job title using the Sentence-Transformer model to obtain an embedding. We then compute the average of these embeddings to produce a single vector representation for the job title query. For example, if the query is "software developer" (which has synonyms like "software engineer", "application programmer", etc. under the same ID), we encode all such terms and average their vectors to capture the broader concept of that job title.

In cases where the query job title might not have multiple variants (or the variants are essentially the same phrase), the single phrase embedding is used as-is. After averaging (if applicable), we apply vector normalization – each query embedding is L2-normalized to unit length. This normalization ensures that similarity computations are cosine-based and not influenced by vector magnitudes. The output of this stage is a set of query embeddings (one vector per query job title).

3.6. Embed Skills

We embed each skill in the skill vocabulary using a similar procedure. For each skill identifier, all alias phrases from the skill's entry are encoded via the Sentence-Transformer model, and their embeddings are averaged to produce one representative vector for that skill. This sentence averaging strategy leverages the multiple descriptions of a skill to capture its full semantic nuance. For instance, if a skill is described by the phrases "pricing strategies", "pricing tactics", and "pricing plans" in the data, we obtain embeddings for each of these and then average them to form a single "pricing strategy skill" vector. As with the queries, we then normalize each skill embedding to unit length. By the end of this stage, we have a library of skill embeddings (each associated with a skill ID) covering the entire skill vocabulary.

3.7. Cosine Similarity Computation

With both query (job title) vectors and skill vectors in the same semantic space, we compute pairwise similarity scores to assess relevance. For each query job title embedding, we calculate its cosine similarity with every skill embedding. Cosine similarity is used as the relevance metric since it effectively measures the angular proximity between the normalized vectors.

Given two unit-length vectors, the cosine similarity is equivalent to their dot product, producing a score in the range [-1, 1] where 1 indicates identical semantics and 0 indicates orthogonality (and negative values indicate opposite semantics). In our context, higher cosine scores indicate that the skill is more semantically related to the job title. This stage takes as input a query vector and iterates over all skill vectors, outputting a list of similarity scores (one score for each skill) for that query.

3.8. Scoring and Ranking

In the ranking stage, the similarity scores generated for each query are used to produce an ordered list of predicted skills. For a given job title query, we sort all candidate skills in descending order of their cosine similarity score relative to the query. This yields a ranked list where the skill with the highest score is deemed the most relevant skill for that job title, the second highest score is the next most relevant, and so on.

We then assign rank positions (starting from 1 for the top skill) to each skill for the query. At this point, we may also apply a cutoff if required by the evaluation (for example, keeping only the top N predictions or all skills above a certain score threshold), although in our implementation we considered the full ranking of skills for completeness. The output of this stage, for each query, is a ranked list of skill IDs accompanied by their similarity scores and rank positions.

3.9. Output – TREC Run File

Finally, we format the ranked results into the standard TREC run file format for submission. The TREC format requires six columns per line: q_id, Q0, doc_id, rank, score, and run_tag. In our case, q_id is the query identifier (the job title ID or query index), and doc_id is the retrieved skill's identifier. We populate each line with the query ID, the placeholder "Q0" (required by the format), the skill ID, the rank (position) of that skill for the query, the similarity score, and a run tag identifying our system.

An example line would be: "q123 Q0 skill456 1 0.873 run1" indicating that for query q123, the top-ranked result is skill456 with a score of 0.873. We include header columns as specified and ensure the file adheres to the exact formatting guidelines of the competition. The output of this stage is the completed run file ready for submission and evaluation.

4. Results and Analysis

Our system's performance on the Task B test set was modest but positive. In the official Codabench leaderboard, our run ranked 19th out of 26 participating systems, narrowly outperforming the organizer's baseline (which ranked 21st) by two positions. This indicates that our multilingual embedding approach with synonym augmentation provided a slight improvement over the baseline. We achieved this improvement without any task-specific supervised training, relying solely on pre-trained mMPNet embeddings and data normalization. However, the gap between our system and the top performers remains significant, suggesting there is ample room for improvement.

To better understand our system's behavior, we conducted an error analysis on its output. Overall, the approach performs well on queries where the job title has an unambiguous, specific skill set associated with it. For instance, given the query "Python Developer", our model correctly ranks "Python" as a top relevant skill, along with other highly relevant technical skills like "Software Development" and "Django". This suggests that the multilingual SBERT embeddings successfully capture strong semantic links between well-known technologies and the job title. In such cases, the cosine similarity between the query and skill embeddings is high for the truly required skills, leading to accurate retrieval.

However, our analysis also revealed several failure patterns. A common issue is the tendency to retrieve very generic skills at high ranks, even when the job query would require more specialized expertise. We observed that broad skills such as "Communication Skills", "Management", or "Microsoft Excel" often appear among the top recommendations for many different job titles. For example, for the query "Stock Broker", the system's top results included "Sales" and "Communication Skills" – skills that are indeed relevant but overly general – while more specific finance-related skills like "Financial Markets" or "Equities Trading" were ranked lower. This bias toward frequent, generic skills is likely due to the embedding model learning that these terms are semantically related to a wide range of jobs, thus yielding high cosine similarity for many queries. Anand et al. (2022) encountered a similar issue: in their results, "Excel" and "Communication Skills" were initially ranked highly for Stock Broker, overshadowing domain-specific skills [10]. They addressed this by multiplying skill scores

Table 2
TalentCLEF Task B Leaderboard on Codabench

#	Participant	MAP
1	pjmathematician	0.360
2	mawhab	0.345
3	pjmathematician	0.342
4	mawhab	0.339
5	pjmathematician	0.333
6	NLPnorth	0.290
7	NLPnorth	0.283
8	iagox	0.278
9	iagox	0.274
10	jensjoris	0.265
11	jensjoris	0.255
12	hallucinators	0.253
13	NLPnorth	0.253
14	iagovazquez	0.250
15	iagox	0.250
16	napatnicky	0.233
17	napatnicky	0.233
18	ahtisham	0.224
19	**COTECMAR-UTB**	**0.215**
20	Beamery Edge	0.205
21	TalentCLEF	0.196
22	srtej	0.141
23	srtej	0.111
24	srtej	0.111
25	Zia	0.035
26	Zia	0.034

with an inverse document frequency (IDF) factor, which dramatically boosted specialized skills like "Securities" and "Financial Services" in the ranking. In our current system, we did not implement such a weighting, which likely explains why generic skills can dominate the results. Incorporating an IDF-based adjustment or other importance weighting could be a straightforward improvement to consider, to penalize skills that are ubiquitous across many job titles.

Another class of errors arises from the synonym expansion strategy. While leveraging jobid2terms and skillid2terms (lists of alternate titles or aliases) generally improved recall, it also introduced noise in some cases. If a job title's expanded terms include very broad or ambiguous words, the averaged query embedding may drift from the core meaning of the job. For example, suppose a query "Marketing Specialist" is expanded with terms like "Marketing Manager" or "Sales Specialist" from the synonyms list; the combined representation might lean toward managerial or sales skills that are not strictly required for the original query. We noticed a few instances where the top-ranked skills seemed more aligned with a synonym or related concept than with the exact query. This points to a limitation of averaging embeddings: irrelevant or loosely related synonyms can dilute the representation. A potential remedy is to weigh synonyms by their relevance or even apply a more sophisticated query expansion technique.

In summary, our error analysis highlights that while the retrieval approach is conceptually straightforward but it struggles with distinguishing skill importance and can be influenced by overly general terms or noisy synonyms. These observations align with findings in related literature and point toward clear directions for improvement. Incorporating techniques from recent research, such as skill frequency weighting, task-specific fine-tuning of the embeddings [10], or even hybrid models that combine text similarity with graph knowledge [12], could substantially enhance both the ranking accuracy and the robustness of our system.

5. Conclusions and future work

The above results and examples underline a key point: our retrieval-based method is easy to implement and language-agnostic, but it does not capture nuances of skill importance or context as well as more sophisticated approaches. The lack of fine-tuning or additional modeling means the system treats the task as pure semantic similarity. This yields reasonable rankings for obvious query-skill matches, but it fails to differentiate which skills are truly "required" versus merely related. By engaging more deeply with related work, we can identify several enhancements. First, moving from a purely unsupervised approach to a weakly-supervised or fine-tuned model would likely yield significant gains, as demonstrated by prior studies. Second, introducing an error-correction mechanism or reranker that accounts for skill frequency could help prioritize niche skills over general ones, similar to how Anand et al. applied an IDF adjustment [10].

In conclusion, our participation showed that a straightforward multilingual embedding approach can serve as a strong baseline, outperforming the provided baseline and validating the feasibility of language agnostic retrieval for this task. At the same time, the gap between our results and the state of the art highlights the importance of incorporating task specific insights and training. Future work should aim to blend the simplicity and coverage of our method with the precision of learned, specialized models for example, by fine-tuning on job-skill pairs, leveraging contrastive learning with domain data, and applying post-processing to address the error patterns identified. This combined strategy could substantially improve the ranking of skills for a given job title, making the system more practical for real-world talent matching applications.

Acknowledgments

We would like to express our sincere gratitude to COTECMAR for providing the necessary space and resources to carry out this research. We also extend our thanks to the Universidad Tecnológica de Bolívar (UTB) for offering the facilities and processing services for running the models, which significantly contributed to the success of this work. Additionally, we acknowledge the support received through the "Convocatoria 950" of 2024 sponsored by Minciencias, which provided the resources for the scholarship that made this research possible.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4, ChatGPT-4o-mini, and o3 AI models to assist with the enhancement and reorganization of the text. These tools were employed to improve pragmatic clarity, optimize overall coherence, and ensure consistency in scientific language. Additionally, DeepL was utilized on certain occasions for translation purposes. Following the use of these tools, the authors reviewed and edited the content as necessary and take full responsibility for the publication's content.

References

[1] G. Gamarra, What is human resources (hr)?: Hr services & responsibilities (2025). URL: https://factorial.es/blog/que-son-recursos-humanos-definicion/#:~:text=Los%20Recursos%20Humanos%20son%20un,%2C%20promoci%C3%B3n%2C%20n%C3%B3minas%20y%20despidos.

[2] I. L. Organization, About the ilo (2025). URL: https://www.ilo.org/about-ilo.

[3] L. Sands, What is human resources (hr)?: Hr services & responsibilities (2023). URL: https://www.breathehr.com/en-gb/blog/topic/business-process/why-is-human-resources-important.

[4] EuropeanCommission, European Skills/Competences, Qualifications and Occupations (ESCO), Technical Report, European Commission, 2024. URL: https://esco.ec.europa.eu/es/about-esco/what-esco.

[5] 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.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need (2017). URL: https://arxiv.org/abs/1706.03762.

[7] D. R, Resume screening using nlp, i-manager's Journal on Information Technology (2024).

[8] J.-J. Decorte, J. V. Hautte, T. Demeester, C. Develder, Jobbert: Understanding job titles through skills (2021). URL: http://arxiv.org/abs/2109.09605.

[9] R. Zbib, L. A. Lacasa, F. Retyk, R. Poves, J. Aizpuru, H. Fabregat, V. Simkus, E. García-Casademont, Learning job titles similarity from noisy skill labels, 2023. URL: http://arxiv.org/abs/2207.00494.

[10] S. Anand, J.-J. Decorte, N. Lowie, Is it required? ranking the skills required for a job-title (2022). URL: http://arxiv.org/abs/2212.08553.

[11] M. Y. Bocharova, E. V. Malakhov, V. I. Mezhuyev, Vacancysbert: the approach for representation of titles and skillsfor semantic similarity search in the recruitment domain, Applied Aspects of Information Technology 6 (2023) 52–59. URL: http://dx.doi.org/10.15276/aait.06.2023.4. doi:10.15276/aait.06.2023.4.

[12] D. Deniz, F. Retyk, L. García-Sardiña, H. Fabregat, L. Gasco, R. Zbib, Combined unsupervised and contrastive learning for multilingual job recommendation, in: M. Kaya, T. Bogers, D. Graus, C. Johnson, J.-J. Decorte, T. D. Bie (Eds.), Recommender Systems for Human Resources, 2024. URL: https://ceur-ws.org/Vol-3788/.

[13] N. Laosaengpha, T. Tativannarat, C. Piansaddhayanon, A. Rutherford, E. Chuangsuwanich, Learning job title representation from job description aggregation network (2024). URL: http://arxiv.org/abs/2406.08055.

[14] H. Fabregat, R. Poves, L. Lacasa, F. Retyk, L. García-Sardiña, R. Zbib, Inductive graph neural network for job-skill framework analysis, Sociedad Española para el Procesamiento del Lenguaje Natural 73 (2024) 83–94. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6602. doi:10.26342/2024-73-6.

[15] R. Alonso, D. Dessí, A. Meloni, D. R. Recupero, A novel approach for job matching and skill recommendation using transformers and the o*net database, Big Data Research 39 (2025). URL: https://www.sciencedirect.com/science/article/pii/S2214579625000048. doi:10.1016/j.bdr.2025.100509.

[16] J. Rosenberger, L. Wolfrum, S. Weinzierl, M. Kraus, P. Zschech, Careerbert: Matching resumes to esco jobs in a shared embedding space for generic job recommendations, Expert Systems with Applications 275 (2025) 127043. URL: https://www.sciencedirect.com/science/article/pii/S0957417425006657. doi:10.1016/J.ESWA.2025.127043.