NT Team at Multilingual Job Title Matching Task A: Job Matching via Large Language Model-Based Description Generation and Retrieval

Ho Thuy Nga[1,2,†], Ho Thi Thanh Tuyen[1,2,†] and Dang Van Thin[1,2,†]

[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]National University, Ho Chi Minh City, Vietnam

Abstract

Job matching is a difficult problem in the labor market due to the vague nature of job titles and the differences in language and industry terms. These challenges make it hard to compare, classify, or retrieve similar job titles in different contexts. The multilingual job title matching task focuses on identifying and ranking job titles that are semantically similar across languages and industries, facilitating more accurate and consistent job classification across languages and domains. To address this task, we propose a system that leverages Large Language Models to enrich job title understanding and enhance matching performance. By combining generative and retrieval-based components, our approach captures semantic relationships, supports multilingual input, and demonstrates adaptability across diverse job domains.

Keywords

Job Matching, Large Language Models, Generative and Retrieval-based Components, Multilingual Input

1. Introduction

In the context of an increasingly dynamic labor market, the precise identification and semantic alignment of job titles play a critical role in a wide range of human resource and talent management tasks, including candidate-job matching, career path modeling, and strategic workforce planning. However, the heterogeneous and constantly evolving nomenclature of job titles presents considerable obstacles for automated systems, which often struggle to accurately interpret, normalize, and link functionally equivalent or related occupational roles. The TalentCLEF Task A Challenge aims to develop systems capable of retrieving and ranking semantically similar job titles from a predefined knowledge base, given an input job title [1]. The task is multilingual in scope, requiring support for English, Spanish, and German, with Chinese as an optional language.

In this paper, we propose a multilingual job title matching system that combines Large Language Model (LLM) with keyword-based and embedding-based retrieval, followed by LLM re-ranking. Given an input title, the system uses an LLM to generate a descriptive representation, computes its embedding, retrieves the top-k most similar titles from a corpus of LLM-generated descriptions using both keyword and embedding similarity, and refines the final ranking via LLM-based semantic re-ranking.

The remainder of the paper is organized as follows. Section 2 provides the related work. The system description is presented in Section 3, followed by results and discussion in Section 4, and Section 5, respectively.

2. Related work

Job title matching has been a long-standing problem in labor market analytics, particularly in job recommendation systems, resume parsing, and occupational classification. Recent advances leverage

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain
*Corresponding author.
†These authors contributed equally.
22520926@gm.uit.edu.vn (H. T. Nga); 22521627@gm.uit.edu.vn (H. T. T. Tuyen); thindv@uit.edu.vn (D. V. Thin)

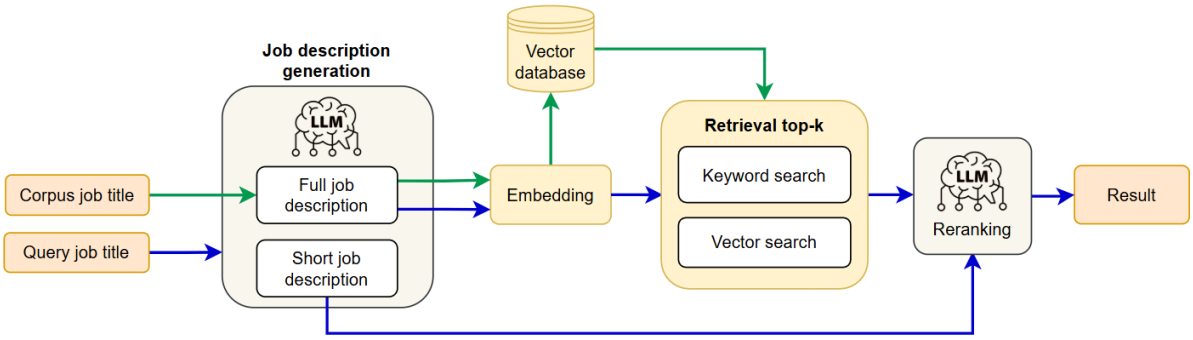

Figure 1: Pipeline for job title matching using Large Language Model generation, hybrid retrieval, and re-ranking.

semantic representation techniques to address the variability and ambiguity of job titles across domains and languages.

Several studies have explored the use of pretrained embeddings for semantic similarity. Building on prior research in labor market analytics, Zhu et al. [2] explored semantic similarity techniques for job title classification, introducing Carotene — a system that leverages Word2Vec-based embeddings and Word Movers Distance to align job titles with standardized taxonomies. Their work demonstrated the effectiveness of pretrained embeddings in addressing the variability of job titles across domains. Expanding on this, Wang et al. [3] proposed DeepCarotene, a multi-stream convolutional neural network that incorporates both word- and character-level representations, significantly improving classification performance over previous models.

Domain adaptation has also been shown to be effective. For instance, ESCOXLM-R [4] adapted the multilingual XLM-R model using masked language modeling and ESCO taxonomical structure, improving performance on job-related classification and sequence labeling tasks across 27 languages. The model particularly excels in capturing short, entity-level spans common in occupation and skill data. Similarly, JobBERT [5] demonstrates that adapting pre-trained language models to the job market domain—by incorporating skill information into job title understanding—can significantly improve performance in occupation classification and job title normalization tasks. This highlights the importance of domain-specific signals in effectively modeling occupation-related semantics.

Addressing the challenge of rare and out-of-vocabulary (OOV) job titles, Ha et al. [6] developed a character-level neural model trained to replicate standard word embeddings. Their method enhanced robustness in job title matching tasks, outperforming traditional embedding techniques and demonstrating effective normalization of rare terms. In multilingual settings, aligning job postings with taxonomies such as ESCO becomes more complex due to language differences. Recent work has shown that multilingual models like XLM-R and large language models (LLMs) can effectively support cross-lingual job matching. Experiments with zero-shot classification and LLM-based annotation demonstrate strong performance in mapping job postings in Italian and Spanish to English ESCO labels [7].

3. Methodology

We participated in Task A: Multilingual Job Title Matching at TalentCLEF 2025, which focuses on identifying and ranking job titles that are semantically similar to a given query title. The task is multilingual, covering English, Spanish, German, and optionally Chinese, and aims to support job matching across different languages and professional domains.

3.1. Dataset

We were provided with a multilingual dataset of job titles in English, Spanish, German and Chinese, collected from diverse job domains. The corpus was designed to support the identification and ranking

Table 1
Comparison of full and short job descriptions

Full job description	Short job description
Recording Engineer A recording engineer specializes in capturing, editing, and mixing audio to produce high-quality recordings, ensuring optimal sound clarity and artistic integrity for music, film, or broadcast projects. **Responsibilities:** Operate and maintain recording equipment (microphones, mixing consoles, DAWs); manage audio quality. **Skills:** Proficiency in DAWs (Pro Tools, Logic Pro), audio signal flow, audio processing tools. **Industry:** Music Production, Film & TV, Broadcast Media.	**Recording Engineer** A recording engineer specializes in capturing, editing, and mixing audio to produce high-quality recordings, ensuring optimal sound clarity and artistic integrity for music, film, or broadcast projects.

of semantically similar job titles across languages and industries.

The training set includes job titles linked to standardized taxonomies such as ISCO (International Standard Classification of Occupations) and ESCO (European Skills, Competences, Qualifications and Occupations), which helps guide model learning across occupational structures. In contrast, the validation and test sets do not contain such taxonomy links, but instead provide query titles, corpus elements, and binary relevance labels annotated by domain experts. These annotations ensure consistent and accurate evaluation across different languages. Participants must generate TREC-formatted ranked lists of similar job titles for each test query. The consistent structure of the validation and test sets enables standard evaluation methods in information retrieval.

3.2. System Description

The architecture of our system designed for Task A: Multilingual Job Title Matching is illustrated in Figure 1. It consists of four main components: Job Description Generation, Embedding, Retrieval, and re-ranking. Each component is described in detail below.

3.2.1. Job Description Generation

Starting from an input job title, a Large Language Model (LLM), specifically Deepseek-chat, is used to automatically generate a detailed job description covering key responsibilities and required skills. From this description, a concise short-form summary is then extracted. An example illustrating this process is shown in Table 1, and the prompt used for guiding the model is provided in Appendix 6.

3.2.2. Embedding

The full job descriptions generated during the Job description generation stage are subsequently processed by alternative text embedding models, including multilingual-e5-large, text-embedding-3-large (OpenAI), and gemini-embedding-exp-03-07 (Google), to convert the textual content into dense vector representations. This embedding process transforms the descriptions into numerical vectors that encapsulate their semantic information, thereby facilitating efficient similarity searches and enabling various downstream analytical tasks.

3.2.3. Retrieval

For the Retrieval stage, the system implements two distinct strategies to identify relevant job descriptions based on a given query. In the first approach, **vector-based retrieval**, the dense vector representations

obtained during the Embedding stage are utilized. Each query is encoded into a semantic vector, and similarity is computed using cosine similarity between the query vector and the job description vectors. The system then retrieves the top 100 most similar job descriptions ranked by cosine similarity scores. In the second approach, **hybrid retrieval**, the system combines both dense semantic similarity and sparse keyword-based relevance to enhance retrieval performance. The final relevance score for each candidate document is computed as a weighted combination of the cosine similarity score from the dense vectors and the keyword matching score from a sparse retrieval method. Formally, the hybrid score is defined as:

$$\text{score} = \alpha \times \text{semantic\_score} + (1 - \alpha) \times \text{keyword\_score}, \tag{1}$$

where $\alpha \in [0, 1]$ controls the balance between the semantic and lexical components. The top 100 results with the highest hybrid scores are selected for further use.

3.2.4. Re-ranking

Finally, re-ranking stage is applied to refine the relevance of the retrieved results. Specifically, the top 100 candidates obtained from the retrieval stage are further processed by a Large Language Model (LLM) to improve ranking quality. For each candidate, the corresponding short-form job description—generated in the Job Description Generation stage—is used as input to the LLM, specifically Deepseek-chat, alongside the original query. The Large Language Model then assigns a relatedness score ranging from 0 to 10, reflecting the semantic alignment between the query and the candidate description. We designed specific prompts to guide the model behavior (Appendix 7). To compute the final ranking score, this LLM-derived score is normalized to the range $[0, 1]$ by dividing it by 10, and then averaged with the original retrieval score (either semantic similarity or hybrid score) as follows:

$$\text{final\_score} = \frac{1}{2}\left(\frac{\text{LLM\_score}}{10} + \text{retrieval\_score}\right) \tag{2}$$

Candidates are then reranked based on this final score, enabling more accurate prioritization of results that are both semantically and contextually aligned with the user's intent.

3.3. Models

We utilized several advanced models in our work, each chosen for its unique capabilities in natural language understanding and generation:

- **deepseek-chat**: deepseek-chat is a state-of-the-art large language model known for its strong performance in complex reasoning and general-purpose text generation tasks. It is developed as part of the DeepSeek LLM project, which focuses on scaling open-source language models guided by scaling laws, with a training dataset of over 2 trillion tokens and advanced fine-tuning techniques [8].
- **gemini-embedding-exp-03-07**: gemini-embedding-exp-03-07 is an experimental version of Google's Gemini model, designed for enhanced contextual understanding and generation across a wide range of domains. Gemini Embedding leverages Gemini's multilingual and code-understanding capabilities to produce generalizable embeddings applicable to various natural language processing tasks [9].
- **text-embedding-large-03-07**: This model is optimized for generating high-quality text embeddings, making it ideal for tasks such as semantic search, clustering, and information retrieval.
- **multilingual-e5-large**: Multilingual-e5-large is a multilingual embedding model capable of handling text in multiple languages, enabling cross-lingual understanding and retrieval tasks effectively [10].

Table 2
Matching results using only job title embeddings

Method	English	German	Spanish	Chinese	Average
multilingual-e5-large	0.3082	0.1821	0.2386	0.3152	0.2610
gemini-embedding-exp-03-07	**0.5935**	0.3564	0.35	**0.5065**	0.4516
text-embedding-large-3	0.5629	**0.3613**	**0.4094**	0.4848	**0.4546**

3.4. Evaluation

The evaluation metric for our job matching system is Mean Average Precision (MAP), which measures the quality of the ranked list of predicted job matches by averaging the precision scores at the ranks where relevant items occur. MAP is computed to assess the system's effectiveness across different language scenarios.

4. Results and Discussion

To evaluate the effectiveness of our proposed system for Task A: Multilingual Job Title Matching, we conduct experiments under three distinct configurations. Each configuration is designed to isolate the contribution of specific components within our architecture. Performance is evaluated using standard information retrieval metrics, with mean average precision (MAP) as the primary evaluation metric, on a multilingual validation and test set.

4.1. Experiment 1: Job Title Embedding Only

In the first experiment, we evaluate a baseline where neither the Job description generation nor the re-ranking module is applied. Instead, only the original job titles are embedded directly using the embedding models described in the system architecture, namely multilingual-e5-large, gemini-embedding-exp-03-07, text-embedding-3-large. Retrieval is conducted using vector-based similarity on these embedded titles. This experiment serves as a lower-bound reference for matching performance, as it captures only the raw semantic information present in the job title without any contextual enrichment.

The results from Experiment 1 in Table 2 confirm the hypothesis that directly embedding raw job titles without additional context yields limited retrieval performance. Among the three models evaluated, `text-embedding-3-large` and `gemini-embedding-exp-03-07` significantly outperform `multilingual-e5-large`, particularly in English and Chinese datasets. This suggests that newer embedding models with broader semantic capacity are better suited for capturing the distinctions among job titles even without context. However, the results indicate that title-only representations are insufficient for capturing the deeper semantics of job roles. These results highlight the necessity of incorporating richer contextual signals — such as job descriptions or re-ranking strategies — for more effective job matching across languages.

4.2. Experiment 2: Full Job Description without re-ranking

The second configuration incorporates the Job description generation module to enrich each job title with detailed textual content. Full-length descriptions generated in Job description generation stage are embedded into dense vectors. Retrieval is performed using both vector-based and hybrid strategies as described previously. However, in this setting, no re-ranking module is applied—the top 100 results are ranked solely based on their semantic or hybrid relevance scores. This experiment demonstrates the impact of LLM-generated descriptions on improving retrieval effectiveness.

Table 3 reports retrieval results under the second configuration, which uses LLM-generated job descriptions without re-ranking. At $\alpha = 0$ (pure semantic search), both embedding models already yield strong performance. For instance, `gemini-embedding-exp-03-07` achieves scores of 0.6749

Table 3
Matching results with full job descriptions without re-ranking, using embedding, hybrid, and keyword retrieval.

Method	Language	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
gemini-embedding-exp-03-07	English	0.6749	0.6796	0.682	0.6843	**0.6851**	0.684	0.6804	0.6749	0.668	0.6576	0.6472
	German	0.4749	0.4772	0.4785	0.4804	0.4821	**0.4827**	0.4824	0.4796	0.4726	0.4576	0.4217
	Spanish	0.4827	0.4886	0.4912	0.4936	**0.4946**	0.4939	0.492	0.4881	0.4827	0.4755	0.4663
	Chinese	0.5961	**0.5984**	0.5973	0.5951	0.5924	0.5867	0.5792	0.5686	0.5578	0.5475	0.5309
	AVG	0.5572	0.561	0.5623	0.5634	**0.5636**	0.5618	0.5585	0.5528	0.5453	0.5346	0.5165
text-embedding-large-3	English	0.6750	0.6799	0.6829	0.6840	**0.6854**	0.6853	0.6812	0.6760	0.6685	0.6590	0.6472
	German	0.4739	0.4766	0.4781	0.4794	0.4813	**0.4826**	0.4822	0.4802	0.4739	0.4593	0.4217
	Spanish	0.5055	0.5088	0.5097	0.5105	**0.5106**	0.5087	0.5041	0.4976	0.4903	0.4802	0.4663
	Chinese	0.6040	**0.6081**	0.6067	0.5977	0.6043	0.5908	0.5814	0.5721	0.5614	0.5485	0.5309
	AVG	0.5646	0.5684	0.5694	0.5679	**0.5704**	0.5669	0.5622	0.5565	0.5485	0.5368	0.5165

Table 4
Matching results with full system on different languages

Method	English	German	Spanish	Chinese	Average
gemini-embedding-exp-03-07	0.6970	0.4812	0.5002	0.6013	0.5699
text-embedding-large-3	0.6974	0.4801	0.5125	0.6089	**0.5748**

(English), 0.4749 (German), 0.4827 (Spanish), and 0.5961 (Chinese), with an average of 0.5572. Similarly, `text-embedding-large-3` performs slightly better with 0.675 (English), 0.4739 (German), 0.5055 (Spanish), and 0.604 (Chinese), averaging 0.5646.

Across all languages, hybrid retrieval consistently outperforms both pure semantic ($\alpha = 0$) and keyword-based retrieval ($\alpha = 1.0$). Both models reach peak performance at $\alpha = 0.4$, indicating that a integration of keyword signals improves results. Notably, performance declines at higher α values, particularly in non-English languages such as Chinese, where keyword-based methods are less effective.

Overall, `text-embedding-large-3` yields stronger results than `gemini-embedding-exp-03-07`, especially in semantic-dominant settings. These findings highlight the value of LLM-generated descriptions and hybrid retrieval strategies for improving multilingual job search performance.

4.3. Experiment 3: Full System

The final experiment evaluates the full system, including the re-ranking module. After generating job descriptions and retrieving the top 100 candidates, we apply an LLM-based re-ranking method using short-form job descriptions. Each candidate is scored for semantic relatedness to the query using Deepseek-chat, and the final score combines the normalized LLM score and the original retrieval score. This setup leverages both rich content generation and advanced contextual re-ranking to optimize match quality. Empirical results show that this configuration achieves the highest performance across all evaluation metrics, validating the effectiveness of integrating LLMs throughout the pipeline.

Table 4 reports the detailed results of this configuration, which employs hybrid retrieval with $\alpha = 0.4$—a value optimized in Experiment 2. Among the evaluated methods, `text-embedding-large-3` achieves the highest average score (0.5748), slightly outperforming `gemini-embedding-exp-03-07` (0.5699). These results further confirm the benefit of incorporating LLM-based re-ranking into the multilingual candidate-job matching pipeline.

4.4. Evaluation on Test Set

Final performance is evaluated on the held-out test set using the same full system setup from validation (Table 5), where the final scores are evaluated by the organizers on CodaBench.

Table 5
Evaluation on Test Set

Method	English	German	Spanish	Chinese	Average
text-embedding-large-3	0.523	0.466	0.404	0.497	0.460

5. Conclusion

In this paper, we present a job matching system developed for Task A of TalentCLEF 2025. Our approach integrates hybrid retrieval methods, along with re-ranking mechanisms powered by large language models. We also leverage LLMs to generate job descriptions for better representation. Experimental results show that combining keyword-based retrieval, semantic embeddings, and LLM-based re-ranking significantly improves the relevance of job-job pairings. Furthermore, exploring more effective prompting strategies for LLMs could further enhance the system's overall performance.

Acknowledgments

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

Declaration on Generative AI

During the preparation of this work, we used GPT-4 in order to: check grammar, spelling, and edit the content for clarity and coherence. After using this tool, we reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] L. Gasco, H. Fabregat, L. García-Sardiña, P. Estrella, D. Deniz, A. Rodrigo, R. Zbib, Overview of the TalentCLEF 2025: Skill and Job Title Intelligence for Human Capital Management, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2025.

[2] Y. Zhu, F. Javed, O. Ozturk, Semantic similarity strategies for job title classification (2016). doi:10.48550/arXiv.1609.06268.

[3] J. Wang, K. Abdelfatah, M. Korayem, J. Balaji, Deepcarotene -job title classification with multi-stream convolutional neural network, in: 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 1953–1961. doi:10.1109/BigData47090.2019.9005673.

[4] M. Zhang, R. Goot, B. Plank, Escoxlm-r: Multilingual taxonomy-driven pre-training for the job market domain, 2023, pp. 11871–11890. doi:10.18653/v1/2023.acl-long.662.

[5] S. Schulz, B. Pelzer, C. Biemann, Jobbert: Understanding job titles through skills (2021). doi:10.48550/arXiv.2109.09605.

[6] P. Ha, S. Zhang, N. Djuric, S. Vucetic, Improving word embeddings through iterative refinement of word- and character-level models, 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. 1204–1213. URL: https://aclanthology.org/2020.coling-main.104/. doi:10.18653/v1/2020.coling-main.104.

[7] H. Kavas, M. Serra-Vidal, L. Wanner, Enhancing job posting classification with multilingual embeddings and large language models, in: F. Dell'Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 440–450. URL: https://aclanthology.org/2024.clicit-1.53/.

[8] X. Bi, D. Chen, G. Chen, et al., Deepseek llm: Scaling open-source language models with longtermism (2024). doi:doi.org/10.48550/arXiv.2401.02954.
[9] J. Lee, F. Chen, S. Dua, et al., Gemini embedding: Generalizable embeddings from gemini (2025). doi:10.48550/arXiv.2503.07891.
[10] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Multilingual e5 text embeddings: A technical report (2024). doi:10.48550/arXiv.2402.05672.

A. Prompt Job Description Generation

Table 6 is the prompt design used to generate detailed and well-structured job descriptions based on the input job title.

Table 6
Prompt Design for Job Description Generation

Generate a structured, professional, and well-formatted job description for the role of `{job_title}`.

Formatting and Guidelines:
– Use clear, concise, and industry-relevant language.
– Avoid company-specific details (e.g., salary, benefits, company history).
– Output must be structured consistently as follows:
1. **Job Title:** `{job_full_description}`
2. **Short Description:** One-sentence summary emphasizing the role's core function.
3. **Responsibilities:** List 4–6 key responsibilities using active verbs (e.g., "Develop", "Manage").
4. **Skills Required:** Group 3–5 core skills into:
   – Technical Skills: languages, tools
   – Soft Skills: communication, leadership
   – Industry Knowledge: standards, compliance
5. **Industry:** List one or more related industries.

Example Output:
Job Title: Software Engineer (Backend)
Short Description: A backend software engineer specializing in building and maintaining scalable server-side applications.
Responsibilities:
– Develop, test, and maintain backend systems using Python and Django.
– Optimize database queries (PostgreSQL, Redis).
– Design and implement RESTful APIs.
– Ensure security best practices.
– Collaborate with frontend engineers and DevOps.
Skills Required:
– **Technical:** Python, Django, REST APIs
– **Databases:** PostgreSQL, MongoDB
– **Cloud/DevOps:** AWS/GCP, CI/CD
– **Soft Skills:** Communication, teamwork
Industry: Software Development | IT Services

B. Prompt re-ranking

Table 7 is the prompt design used for the re-ranking task, where the model assesses the semantic similarity between two short job descriptions.

Table 7
Prompt Design for re-ranking

Given two short job descriptions written in English, evaluate how related they are on a scale from 0 to 10. Think step by step before giving your final score.

Follow this reasoning process:
– Compare job titles
– Compare domains
– Compare job responsibilities

Scoring scale (based on your reasoning):
– 10: Almost the same job (same title, industry and role)
– 8–9: Very closely related (similar work, same domain, strong role overlap)
– 6–7: Related in the same field with some overlap in tasks
– 3–5: Slightly related (same industry but different roles)
– 0–2: Barely related

Think carefully through each aspect above and then provide **only the final score**, from 0 to 10, with no explanation.

Input Format:
– Job 1: `{short_description_job1}`
– Job 2: `{short_description_job2}`