Enterprise Candidate-Job Matching with Large Language Models

Anonymous ACL submission

Abstract

We explore how to effectively build an enterprise candidate-job matching system, using Language Model Models (LLMs), to enhance the alignment of talents with job requirements. We focus on evaluating the performance of two widely-used LLMs within our proposed "Read-Rewrite-Retrieve-Read" information retrieval (LLM4IR) framework, to find a balance between performance, cost and privacy concerns while automating candidate-job matching in high-volume job application scenarios. We analyse approximately 1.1 million unique CVs submitted in response to 10,411 unique job postings, covering 4,832 distinct job titles. Overall, our results underscore the significance of (i) creating high-quality documents from the CVs that emphasize skills, knowledge, and competencies for accurate matching, while addressing privacy concerns, (ii) augmenting job queries with occupation-related context to enhance matching accuracy, and (iii) identifying an effective method for prompt-tuning specific LLMs, such as retrieval-augmentation for LLaMA-3.1 and few-shot learning for GPT-4o-mini, which is essential for optimising their performance. In practical terms, open-source LLMs can be hosted locally for offline tasks such as information extraction and parsing to achieve the performance-cost-privacy trade-off.

1 Introduction

Recruitment plays a critical role in connecting job seekers with employers in the labour market. Recently, there has been a rise in the use of AI-enabled tools at every stage of the recruitment process, including sourcing, screening, interviewing, and selection. Applicant Tracking Systems (ATS) and Customer Relationship Management (CRM) systems have evolved to merge into intelligent recruitment and talent management platforms. These platforms such as LinkedIn Talent Solutions [1] can automate the generation of job descriptions and postings on multiple job boards, analyse application documents at scale and speed, create better candidate profiles for job matching and recommendations, and enable efficient client and candidate management. By leveraging the power of AI, the hiring process is significantly accelerated compared to traditional methods, with the possibility of describing, posting, and filling a job within a few days.

At the heart of these intelligent recruitment platforms are candidate-job matching algorithms, which facilitate semantic similarity searches to identify the most suitable candidates from a diverse pool of CVs. Recruiters can search application archives using essential job requirements, such as licence names, certifications or entire job ads, resulting in a ranked list of relevant candidates and a sound rationale for the candidate selection. This approach sets a high standard in the initial sourcing stage, broadening access to talent pipelines, targeting candidates with specific skill sets, and reducing operational costs. Moreover, it allows recruiters to focus on more people-involved tasks promising greater efficiency and scalability.

Part of the task of assessing a CV's fit to a written set of job requirements can be performed by looking for semantic similarities between the application documents (such as CVs) to the job requirements. Currently, there are numerous pretrained embedding models that produce contextual text embeddings to measure Semantic

[1] https://business.linkedin.com/talent-solutions

Textual Similarity (STS) between texts (Chandrasekaran & Mago, 2021; Muennighoff et al., 2023). STS methods typically output a similarity or distance measure, such as cosine or Euclidean distance, indicating the level of similarity between the texts. However, beyond ascertaining semantic similarity between a job description and a candidate application, good candidate-job matching algorithms must evaluate suitability of applicants to a specific job description and rank CVs accordingly (Guo et al., 2016; Rezaeipourfarsangi & Milios, 2023; Rojas-Galeano et al., 2022).

Recent improvements in information retrieval (IR) systems integrate Large Language Models (LLMs) in dynamic frameworks coined *LLM4IR*[3] to improve accuracy and user-friendliness. The frameworks leverage the language comprehension, generation, and reasoning capabilities of LLMs to improve retrieval outcomes through pre-retrieval modules for query rewriting, as well as post-retrieval modules for document reranking and result summarization (Ai et al., 2023; Hersh, 2024; X. Li et al., 2024; Xiong et al., 2024; Zhu et al., 2023; Ziems et al., 2023). Consequently, modern IR systems benefit from user intent capture, retrieval based on embeddings and words, multi-stage reranking, and summarization. Notable LLM4IR frameworks include "Retrieve-then-Read" (Shen, Geng, et al., 2023), "Rewrite-Retrieve-Read" (Ma et al., 2023), and "Rewrite-Retrieve-Rerank-Read" (Zhu et al., 2023).

In our study, we utilise GPT-4o-mini (OpenAI et al., 2023) and LLaMA-3.1-8B-Instruct (Llama Team, 2024; Touvron et al., 2023) within our proposed "Read-Rewrite-Retrieve-Read" LLM4IR framework. Specifically, we implement the Document Reader, Query Rewriter and Retriever modules in the framework to align with candidate-job matching for improved outcomes. We provide a performance analysis of how integrating the LLMs in our proposed framework improved the matching accuracy between CVs and job descriptions at scale within a tailored corporate recruitment platform.

2 Related Works

Novel methods for automating candidate-job matching (i.e. matching suitable CVs to job vacancies) primarily focus on two main trajectories for extracting and encoding essential information: (i) utilisation of machine learning algorithms and natural language processing (NLP) techniques, or (ii) fine-tuning pretrained language model to incorporate domain-specific knowledge for improved performance. Typically, essential information extracted from job postings and CVs encompasses three key areas: skills, knowledge, and competencies (SKC). This is in line with the growing importance of skills-based hiring in the staffing industry (Fuller et al., 2022), which focuses on matching candidates to roles based primarily on their SKC profile, rather than past experience.

In the case of the latter approach, despite the potential for improved performance by RecruitPro (Fang et al., 2023), InEXIT (Shao et al., 2023), and the potential of combining language models and Siamese architecture (Özlü et al., 2023; Pezeshkpour et al., 2023; Rezaeipourfarsangi & Milios, 2023; Tanberk et al., 2023; Vanetik & Kogan, 2023; Y. Zhang et al., 2023), it is usually not accessible in commercial settings. This is because the process of integrating such models into enterprise search workflow can be daunting except when they are served via an API as in SkillGPT (N. Li et al., 2023) or from service providers like Lightcast[4] and LinkedIn[5].

The former approach focuses on improving candidate-job matching by analysing skills from online job postings like Sockit (Dixon et al., 2023), SkillSpan (M. Zhang et al., 2022), and maintaining dynamic national occupation-skill profiles as in O*Net[6] and ESCO[7]. Skills2Job (Giabelli et al., 2021), JobBERT (Decorte et al., 2021), GSL (Zheng et al., 2023), (Kara et al., 2023) and person-job fit analysis (He et al., 2023; Yao et al., 2022; Y. Zhang et al., 2023) leverage SKC information to improve on candidate-job matching.

[3] LLM4IR connotes enhancing IR systems with LLMs

[4] https://lightcast.io

[5] https://developer.linkedin.com/product-catalog/talent

[6] https://www.onetonline.org

[7] https://esco.ec.europa.eu/en/classification/occupation_main

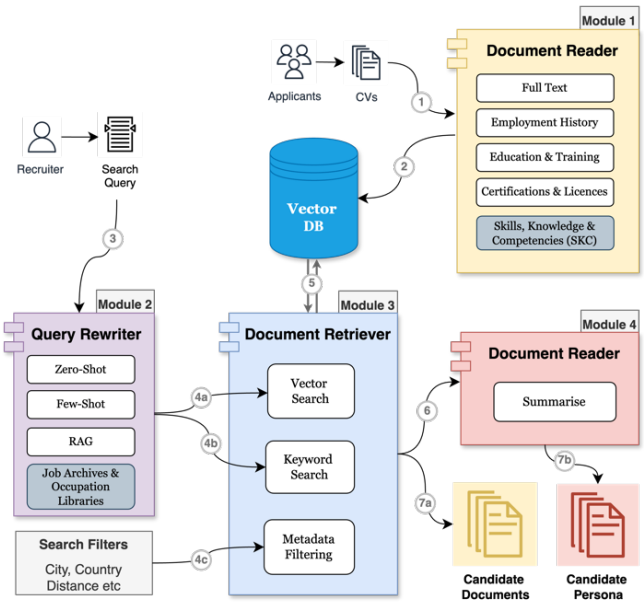

Figure 1: Our proposed "Read-Rewrite-Retrieve-Read" LLM4IR framework for Enterprise Candidate-Job Matching. Modules 1, 2 & 4 use LLM to extract SKC from CVs, rewrite job queries, and summarise retrieved CVs respectively. Module 3 is the vector-based document retriever. (1)-Read CVs; (2)-Index and Store CV embeddings; (3)-Rewrite job query; (4)-Retrieve suitable CVs; (5)-Query vector database; (6)-Summarise CV; (7)-List candidate CVs or persona

Specifically, (Tanberk et al., 2023) utilised NLP and AI models to analyse skills and rank CVs based on their similarity to job descriptions, (Vanetik & Kogan, 2023) annotated a CV dataset with keywords, and named entities for SKC and proposed an unsupervised method for candidate-job ranking. AnnoJob (Brek & Boufaida, 2023) utilised semantic annotation to extract contextual skill entities from CVs and job offers and match them based on semantic similarity; and (Rezaeipourfarsangi & Milios, 2023) conducted a comparative analysis of different models for matching job descriptions with CVs, and demonstrated the potential of combining language models, CNN, LSTM and Siamese architecture to extract and emphasize specific skills as domain-specific knowledge.

In our research, we investigated how to use LLMs to combine these approaches: extract and encode essential entities from CVs as well as incorporate domain-specific knowledge for better talent outcomes. To achieve this, we assess our proposed "Read-Rewrite-Retrieve-Read" LLM4IR framework with the full CV and text extracted from the SKC sections of the CVs. The SKC sections are employment, education, training, certifications and licensing details. We address the questions:

I. Is the entire CV required or can an SKC-centric extraction process capture enough information to meet or exceed the performance achieved by the entire CV?
II. Is additional domain-specific knowledge required to augment job queries to improve candidate-job matching accuracy? What is the most effective technique required to achieve this?

3 Methodology

In this study, we introduce a "Read-Rewrite-Retrieve-Read" LLM4IR framework designed to enhance the alignment of talent with job requirements in practical business contexts. The system design for our proposed framework is depicted in Figure 1. Our approach involves the use of LLMs in both the pre- and post-retrieval modules for job query rewriting, reading and summarising CVs, to enhance the performance of our foundational vector-based document retriever. We investigate methods of integrating LLMs into this enterprise candidate-job matching system to optimise the trade-off between performance and cost.

In the framework, the LLM is employed as a **Document Reader** to extract vital information from CVs. The goal here is to ensure textual information in CVs is appropriately captured to improve the performance of the retrievers. LLMs usually exhibit proficiency in parsing and can generate structured document objects containing comprehensive details such as work history, educational background, professional licenses, and certifications. We utilise the LLM for document reading in two distinct modes: as a pre-retrieval *(Module 1)* for generating high-quality documents and highlighting key skills and competencies from CVs, and as an optional post-retrieval *(Module 4)* for creating candidate profiles or summarising CVs for other downstream recruitment tasks.

Next, the LLM is used as a **Query Rewriter** *(Module 2)* to improve job descriptions and align them more closely with standard role descriptions. This process addressed issues related to short and imprecise queries, typically ranging from job-title phrases to a few key job requirements, often totalling less than 500 characters. Notable approaches to leveraging LLMs for query rewriting include zero-prompt tuning, such as in PromptReps (Zhuang et al., 2024) and HyDE (Gao et al., 2023); few-prompt tuning, as demonstrated in Query2Doc (Wang et al., 2023); and retrieval augmentation (Shen, Long, et al., 2023; Ma et al., 2023; Y. Wang et al., 2023). In this module, we generate augmented job queries by, drawing from LLM intrinsic knowledge, previous and similar job role descriptions, and standard occupation libraries. These methods are assessed to ascertain the most effective approach for generating a comprehensive and detailed description of the role, to enhance talent outcomes.

The **Document Retriever** *(Module 3)* employs vector search based on semantic similarities, 1:1 keyword matching, and document metadata filtering. The document embeddings capture the semantic representation of the CVs allowing for a fast inner-product computation of the query and document vectors and returning a ranked list of CVs with their corresponding distance scores. The documents are indexed and stored alongside their semantic embeddings in a vector database for subsequent hierarchical retrieval. Our choice of the vector database was based on multivariate factors such as speed of retrieval, and metadata filtering to eliminate incorporating basic one-to-one keyword filtering (in the cases of domain-specific requirements like practitioner licences or certifications). These document retriever features are usually supported by popular vector databases such as Pinecone [8], Chroma [9], and traditional PostgreSQL[10] via the pgvector extension[11].

Industry Sector	#Job Ad	# CVs
Manufacturing and Production	1,295	92,885
Logistics Distribution and Supply Chain	1,200	105,320
Education and Training	945	27,681
Admin and Secretarial	827	38,186
Engineering	621	43,011
Call Centre and Customer Service	601	47,353
Medical and Nursing	408	45,820
Retail	395	49,492
Building and Construction	600	67,207
Sales and Marketing	256	21,873
Accountancy	353	6,949
Hospitality and Catering	312	47,836
Utilities	290	51,005
Transport and Rail	239	43,412
Other Sectors	2069	58,186
Total	**10,411**	**1,092,765**

Table 1: Distribution of unique job postings and application CVs across 60 industry sectors. CVs refer to both CVs and resumes.

4 Experiments

4.1 Models

We conduct a performance evaluation of two contemporary LLMs for our enterprise candidate-job matching system. The selected models are notable for their balanced emphasis on efficiency and cost-effectiveness:

GPT-4o-mini is a smaller, more affordable version of the powerful GPT-4o language model. It offers

[8] https://github.com/pinecone-io/pinecone-python-client

[9] https://github.com/chroma-core/chroma

[10] https://www.postgresql.org

[11] https://github.com/pgvector/pgvector

advanced textual intelligence and multimodal reasoning, excelling in comprehending and generating human-quality text. It supports text and image inputs, with a context window of 128,000 tokens, an output capacity of 16,384 tokens, and training data up to October 2023 (OpenAI, 2024).

LLaMA-3.1-8B-Instruct, developed by Meta AI, is known for its exceptional performance across a wide range of natural language tasks. It performs well in understanding and carrying out instructions. Ethical considerations behind its design help to address concerns related to bias, misinformation, and potential misuse. It has a context window of 128,000 tokens, an output capacity of 4,096 tokens, and training data up to December 2023 (Meta-Llama, 2024).

4.2 Job Postings and CVs

Table 1 contains a comprehensive summary of the recruitment data analysed in this study. It includes 10,411 unique job postings for 4,832 distinct job titles, and approximately 1.1 million unique CVs of candidates who responded to job advertisements. These job postings cover over 60 industry sectors in the United Kingdom and were collected in a commercial setting over a 9-month period in 2023. The average number of unique applications per job ranges from 55 to 1,064, depending on the type and location of the role.

4.3 Experimental Setup

We evaluate the performance of our enterprise candidate-job matching system using job postings and their augmented versions as queries. We tested hese against two different CV formats: the full CV and the extracted SKC text. The experimental setup is described as follows:

Document Reading: We employ a hybrid approach to extract high-quality text from the CVs, often submitted in PDF or DOCX formats with diverse text and image layouts. The full CV text is extracted with open-source document readers, python-docx[18] and pypdf[19] Python packages, while all personal and sensitive information was anonymised with MS Presidio Anonymiser[20]. The LLM was prompted to parse the CVs into structured JSON format, extracting the employment, education, training and certification, and licensing sections. Each section included details like name, dates, and description/responsibilities representing the SKC text. Similarly in the post-retrieval document reading, the LLM was prompted to generate a summary of the CV, highlighting how it aligns with the job requirements accordingly.

Query Rewriting: We generate four additional job queries using prompt-tuning and retrieval-augmented approaches, as follows:

i) *LLM-zero-shot*: leveraging the intrinsic knowledge of LLMs from its diverse training data to add more information to the job query.
ii) *LLM-few-shot*: incorporating a few examples of detailed job descriptions into the prompt for in-context learning. The job descriptions captured explicit information on the summary, regular tasks, detailed work activities, technology skills, common tools used, knowledge, and education required for the role from the O*Net occupation-skill library.
iii) *Job-Postings as-a-retriever*: retrieval augmentation of job queries with domain-specific information from past and similar job postings.
iv) *ESCO as-a-retriever*: retrieval augmentation of job queries with domain-specific information from similar ESCO national occupation profiles.

We implement embedding-based retrievers for the occupational profiles obtained from ESCO and past job postings for retrieval augmented generation. The top-5 results were used as context for the LLM to rewrite the queries.

Document Retrieval: We implemented four embedding-based retrievers in this study with OpenAI embedding models[21] and Chroma vector databases. The full CV text and extracted SKC text were encoded into two separate databases with the OpenAI *text-embedding-large-3* model, while the occupational profiles obtained from ESCO, and past job postings were encoded with the OpenAI *text-embedding-small-3* model. Document

[18] https://github.com/python-openxml/python-docx
[19] https://github.com/py-pdf/pypdf
[20] https://github.com/microsoft/presidio
[21] https://platform.openai.com/docs/guides/embeddings

Document Type →	Full CV Text						Extracted SKC Text					
Query Rewriter ↓	Recall		Precision		nDCG		Recall		Precision		nDCG	
	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20	@10	@20
LLM = None												
User Query *Baseline*	0.276	0.150	0.176	0.127	0.892	0.879	0.415	0.446	0.833	0.616	0.955	0.954
LLM = GPT-4o-mini												
LLM- *zero-shot*	0.218	0.773	0.245	0.248	0.893	0.880	0.327	0.401	0.880	0.853	0.971	0.970
LLM- *few-shot*	0.164	0.247	0.318	0.307	0.927	0.912	0.553	0.564	0.803	0.795	0.964	0.965
Job Postings *as-a-retriever*	0.279	0.889	0.171	0.229	0.832	0.812	0.297	0.417	0.859	0.707	0.964	0.965
ESCO library *as-a-retriever*	0.264	0.460	0.175	0.255	0.893	0.880	0.237	0.636	0.689	0.684	0.952	0.949
LLM = LLaMA-3.1-8B-Instruct												
LLM *zero-shot*	0.204	0.506	0.250	0.159	0.832	0.812	0.303	0.417	0.836	0.646	0.983	0.984
LLM *few-shot*	0.150	0.895	0.311	0.318	0.952	0.941	0.306	0.453	0.710	0.711	0.964	0.965
Job Postings *as-a-retriever*	0.417	0.885	0.194	0.256	0.856	0.831	0.307	0.420	0.754	0.688	0.964	0.965
ESCO library *as-a-retriever*	0.284	0.195	0.284	0.309	0.863	0.846	0.331	0.500	0.755	0.671	0.973	0.972

Table 2: Evaluation Results of our proposed "Read-Rewrite-Retrieve-Read" *LLM4IR* framework for Enterprise Candidate-Job matching.

embeddings were compared with cosine distance to determine semantic similarity. The output of our foundational vector-based document retriever is a ranked list of suitable candidates.

5 Results and Discussion

5.1 Evaluation Metrics

In evaluating performance, we use occupation profiles sourced from O*Net as the ground truth for job queries. Occupational information was aligned and translated with the ESCO-O*Net crosswalk (ESCO Publications, 2022) to identify the most appropriate description for each role. We extracted specific fields from the comprehensive O*Net occupation profiles to generate the ground truth: description, tasks, detailed work activities, tech skills, tools used, knowledge, and education.

The evaluation results are detailed in Table 2. The Extracted SKC Text column contains the results for our proposed "Read-Rewrite-Retrieve-Read" LLM4IR framework. The results in the Full CV Text column correspond to the "Rewrite-Retrieve-Read" LLM4IR framework (Ma et al., 2023), as the original CV text is utilised directly in this case. Similarly, the results in the User Query row, under the Full CV Text column, correspond to

Document Type →	Full CV Text		Extracted SKC Text	
Query Rewriter ↓	F1 score		F1 score	
	@10	@20	@10	@20
LLM = None				
User Query *Baseline*	0.215	0.138	0.554	0.517
LLM = GPT-4o-mini				
LLM *zero-shot*	0.231	0.376	0.477	0.546
LLM *few-shot*	0.216	0.274	0.655	0.660
Job Postings *as-a-retriever*	0.212	0.364	0.441	0.525
ESCO library *as-a-retriever*	0.210	0.328	0.353	0.659
LLM = LLaMA-3.1-8B-Instruct				
LLM *zero-shot*	0.225	0.242	0.445	0.507
LLM *few-shot*	0.202	0.469	0.428	0.553
Job Postings *as-a-retriever*	0.265	0.397	0.436	0.522
ESCO library *as-a-retriever*	0.284	0.239	0.460	0.573

Table 3: Enterprise Candidate-Job Matching Accuracy.

	Metrics	Query Rewriting Method				CV Reading****		
		llm_zshot	llm_fshot	job_retriever	esco_retriever	Resume1	Resume2	Resume3
JobQuery/CV	Word Count*	495	495	495	495	500	800	1306
Prompts	Word Count	761	1854	3309	1495	728	1008	1514
GPT-4o-mini (closed-source LLM)	Input tokens	970	3588	3969	2464	1166	1534	2151
	Output tokens	862	841	750	740	1663	2057	2764
	Inference cost/1K ($)	0.66	1.04	1.05	0.81	1.25	1.54	2.07
LLaMA-3.1-8B (open-source LLM)	Input token	969	3632	3981	2457	1175	1542	2165
	Output token	796	614	947	700	1249	2016	1724
	Inference Cost/1K ($)	0.78	1.46	1.77	1.16	1.11	1.69	1.70

Job Descriptions average 500 words
***GPT-4o-mini pricing: $0.15 / 1M input tokens and $0.60 / 1M output tokens (Source: OpenAI Pricing)*
****LLaMA-3.1-8B API endpoint pricing: $0.30 / 1M input tokens and $0.61 / 1M output tokens (Source: Azure marketplace)*
****Resume sampled within the 25th, 50th and 75th percentiles of the wordcount.

Table 4: Inference Cost for Query Rewriting and CV Reading for the "Read-Rewrite-Retrieve-Read" LLM4IR framework for Enterprise Candidate-job matching

the "Retrieve-then-Read" framework (Shen, Geng, et al., 2023).

To assess the semantic overlap between the retrieved candidate CVs and the ground truth, we report IR metrics, Recall (Recall@K), Precision (Precision@K) at top-K where $K = [10,20]$ and the Normalised Discounted Cumulative Gain (NDCG) for ranking accuracy. The Recall@K and Precision@K are computed at sentence/entity level with the ragas[24] python library using GPT-4o as the critic LLM. The matching accuracy was determined using the F1 score, presented in Table 3. In Table 4, we outline the average inference costs for each query rewriting method and CV reading.

5.2 Observations

Shape of the CV: A noticeable improvement in matching accuracy (at least 23%) was achieved when a document reader highlighted essential SKC information from the CVs before implementing the embedding-based retriever. This means the quality of talent-skill matches improves when the contents of the CVs are closely aligned to the job requirements. This corresponds with results from previous studies which emphasise feature engineering around SKC information (Pezeshkpour et al., 2023; Tanberk et al., 2023; Yao et al., 2022). In contrast to other LLM4IR frameworks, such as scenarios involving original user queries ("Retrieve-then-Read") and augmented job queries ("Rewrite-Retrieve-Read"), the inclusion of a pre-retrieval "Read" module enhanced effectiveness in the context of candidate-job matching.

Query rewriting: Generally, augmenting job descriptions with additional occupational context improves the matching accuracy. However, the query rewriting techniques for optimising the performance of LLMs differ. Zero- or few-shot prompt-tuning appears to be more suitable for GPT-4o-mini, yielding a 1.6% and 10.1% increase in accuracy for full CV and SKC text respectively. Conversely, retrieval augmented generation proves to be effective for LLaMA-3.1-8B, resulting in a 6.9% increase in accuracy for full CV text. As the value of K increases, there is an improvement in matching accuracy with augmented job descriptions, but a decrease in accuracy with the user query. This indicates that more suitable candidates are ranked relatively higher in the overall retrieved list, even when derived from a pool of qualified candidates.

Performance: Precision indicates proportion of the 'true entities' in the ground truth (compared

24 https://github.com/explodinggradients/ragas

with the query) found in each ranked CV. Higher precision values indicate the good resumes are ranked higher. The use of LLMs for query rewriting results in improved precision, particularly with standard job descriptor examples GPT-4o-mini and LLaMA-3.1-8B demonstrated notable improvements in precision, with GPT-4o-mini yielding a 14.2% and 4.7% increase in precision for full CV and SKC text respectively, while LLaMA-3.1-8B also yielded a 13.5% and 0.3% increase in precision for full CV and SKC text. These results represent a substantial advancement over the baseline. This implies the cost of implementing an embedding-based retriever to augment job queries can be avoided. Even though the document recall with queries augmented with past or similar job postings is higher with both LLMs, the top-ranked CVs do not closely align with the expected job requirements in the ground truth or job descriptors.

Performance-Cost Trade-off and Privacy Concerns: The implementation of RAG for query rewriting introduces additional costs when compared with the costs of extra tokens with few-shot learning. Moreover, using LLMs with zero or few-shots, can result in more stable and significant outcomes even as K increases. However, a good balance between performance, cost, and privacy therefore might be hosting open-source LLMs locally for offline tasks such as information extraction and parsing. This also without raising privacy concerns.

5.3 A Commercial Perspective

One of the main goals of a recruitment agency is to streamline the registration and onboarding process for low- to medium-skilled temporary workers within 24 hours, reducing time-to-hire for recruiters and employers. The impact of broadening talent pools based on related skills is significant, and LLMs are proving to be instrumental in identifying talent that is an excellent match for our clients from a wider pool. By automating routine recruitment tasks, recruiters can free up to focus on more meaningful interactions with clients and candidates.

An intelligent recruitment platform includes candidate and client portals, which offer onboarding and temporary worker management. It also integrates LLMs to help find talent faster than the market average, with a booking secured in just 23 minutes recently. It is anticipated that AI-generated CVs and cover letters will make recruiters even more important as businesses start to outsource their hiring processes to agencies.

Therefore, it is important to ensure the responsible and transparent use of LLMs and AI tools to handle applicants, as manual sifting becomes more challenging in a crowded market.

6 Conclusions

In this paper, we explored how to effectively build an enterprise candidate-job matching system, using Language Model Models (LLMs), to enhance the alignment of talents with job requirements. This candidate-job matching is a core part of intelligent recruitment systems. We employ the LLMs, GPT-4o-mini and LLaMA-3.1-8B-Instruct, for information extraction, query rewriting, and summarisation tasks within our proposed "Read-Rewrite-Retrieve-Read" LLM4IR framework for candidate-job matching. We observed that prompt-tuning GPT-4o-mini with zero- or few-shot examples from standard occupation profiles to rewrite detailed job queries improves matching accuracy. Conversely, LLaMA-3.1-8B benefitted more from retrieval-augmented query rewriting to enhance matching accuracy. Overall, the results underscore the significance of implementing a document reader to emphasise skills, knowledge, and competencies information from the CVs before retriever. This improves the quality of talent-skill matches since the contents of the read CVs can be closely aligned to the job requirements. Additionally, for optimal performance-cost trade-offs, the open-source LLMs such as LLaMA-3.1-8B can be deployed for offline tasks such as information extraction, parsing, and summarisation. This research thereby provides crucial insights and a clear perspective on deploying LLMs for candidate-job matching in real-world business settings. By automating routine recruitment tasks, we can free up recruiters to focus on more meaningful interactions with clients and candidates.

Limitations

The purpose of this study was to assess the effectiveness of LLM in matching CVs to job openings, which is an essential part of our candidate search process in a specialised recruitment platform. As a result, we exclusively focused on using GPT-4o-mini and LLaMA-3.1-

8B-Instruct as the benchmark in commercial environments. Also, for objective analysis, we considered the comprehensive standard occupation profiles obtained from O*Net as ground truth for job descriptors. While other LLMs and occupation libraries might yield superior results, evaluating them was outside the scope of this study.

Ethics Statement

License: We maintained the licensing requirements accordingly while using different tools from the OpenAI.

Privacy: To protect user privacy, sensitive data such as personally identifiable information (e.g., gender, ethnicity, phone number, person names and emails, URLs) were removed from the data while evaluating the various information retrieval approaches. All possible identifiable materials were also removed to ensure we uphold the tenants of the proprietary data.

Intended Use: This evaluation was carried out to find a balance between performance, cost and privacy concerns while leveraging the current market offerings of LLMs for quicker market deployment of products and services. While the LLMs offer a significant advantage, we intend to protect the interest of our clientele base while being transparent about the use of AI for recruitment tasks. This is however no attempt to engage AI-powered mechanisms that are to the detriment of any demographic group. Only the clients and customers that use our platform will have access to the services.

References

Ai, Q., Bai, T., Cao, Z., Chang, Y., Chen, J., Chen, Z., Cheng, Z., Dong, S., Dou, Z., Feng, F., Gao, S., Guo, J., He, X., Lan, Y., Li, C., Liu, Y., Lyu, Z., Ma, W., Ma, J., … Zhu, X. (2023). Information Retrieval meets Large Language Models: A strategic report from Chinese IR community. *AI Open*, *4*. https://doi.org/10.1016/j.aiopen.2023.08.001

Brek, A., & Boufaida, Z. (2023). AnnoJOB: Semantic Annotation-Based System for Job Recommendation. *Acta Informatica Pragensia*, *12*(2). https://doi.org/10.18267/j.aip.204

Chandrasekaran, D., & Mago, V. (2021). Evolution of Semantic Similarity—A Survey. *ACM Comput. Surv*, *54*, 41–41. https://doi.org/10.1145/3440755

Decorte, J.-J., Van Hautte, J., Demeester, T., & Develder, C. (2021). *JobBERT: Understanding Job Titles through Skills*. http://arxiv.org/abs/2109.09605

Dixon, N., Goggins, M., Ho, E., Howison, M., Long, J., Northcott, E., Shen, K., & Yeats, C. (2023). Occupational models from 42 million unstructured job postings. *Patterns*, *4*(7). https://doi.org/10.1016/j.patter.2023.100757

ESCO Publications. (2022). *The crosswalk between ESCO and O*Net* [Technical Report]. European Commission. https://esco.ec.europa.eu/en/about-esco/publications/publication/crosswalk-between-esco-and-onet-technical-report

Fang, C., Qin, C., Zhang, Q., Yao, K., Zhang, J., Zhu, H., Zhuang, F., & Xiong, H. (2023). *RecruitPro: A Pretrained Language Model with Skill-Aware Prompt Learning for Intelligent Recruitment*. 3991–4002. https://doi.org/10.1145/3580305.3599894

Fuller, J., Langer, C., & Sigelman, M. (2022, February 11). Skills-Based Hiring Is on the Rise. *Harvard Business Review*. https://hbr.org/2022/02/skills-based-hiring-is-on-the-rise

Gao, L., Ma, X., Lin, J., & Callan, J. (2023). *Precise Zero-Shot Dense Retrieval without Relevance Labels*. *1*. https://doi.org/10.18653/v1/2023.acl-long.99

Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., & Seveso, A. (2021). Skills2Job: A recommender system that encodes job offer embeddings on graph databases. *Applied Soft Computing*, *101*. https://doi.org/10.1016/j.asoc.2020.107049

Guo, S., Alamudun, F., & Hammond, T. (2016). RésuMatcher: A personalized résumé-job matching system. *Expert Systems with Applications*, *60*, 169–182. https://doi.org/10.1016/j.eswa.2016.04.013

He, M., Shen, D., Wang, T., Zhao, H., Zhang, Z., & He, R. (2023). Self-Attentional Multi-Field Features Representation and Interaction Learning for Person-Job Fit. *IEEE Transactions on Computational Social Systems*, *10*(1). https://doi.org/10.1109/TCSS.2021.3134458

Hersh, W. (2024). Search still matters: Information retrieval in the era of generative AI. *Journal of the American Medical Informatics Association*. https://doi.org/10.1093/jamia/ocae014

Kara, A., Daniş, F. S., Orman, G. K., Turhan, S. N., & Özlü, A. (2023). *Job Recommendation Based on Extracted Skill Embeddings*. *544 LNNS*, 497–

507. https://doi.org/10.1007/978-3-031-16075-2_35

Li, N., Kang, B., & De Bie, T. (2023). *SkillGPT: a RESTful API service for skill extraction and standardization using a Large Language Model*. http://arxiv.org/abs/2304.11060

Li, X., Jin, J., Zhou, Y., Zhang, Y., Zhang, P., Zhu, Y., & Dou, Z. (2024). *From Matching to Generation: A Survey on Generative Information Retrieval*. http://arxiv.org/abs/2404.14851

Llama Team. (2024). *The Llama 3 Herd of Models | Research—AI at Meta*. https://ai.meta.com/research/publications/the-llama-3-herd-of-models/

Ma, X., Gong, Y., He, P., Zhao, H., & Duan, N. (2023). *Query Rewriting for Retrieval-Augmented Large Language Models*. EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings. https://doi.org/10.18653/v1/2023.emnlp-main.322

Meta-Llama. (2024, July). *Llama 3.1 Model Information*. GitHub. https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md

Muennighoff, N., Tazi, N., Magne, L., & Reimers, N. (2023). MTEB: Massive Text Embedding Benchmark. *arXiv Preprint*. https://arxiv.org/abs/2210.07316

OpenAI. (2024, July). *GPT-4o mini: Advancing cost-efficient intelligence*. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., … Zoph, B. (2023). *GPT-4 Technical Report*. http://arxiv.org/abs/2303.08774

Özlü, A., Orman, G. K., Daniş, F. S., Turhan, S. N., Kara, K. C., & Yücel, T. A. (2023). *Similarity-Based Résumé Matching via Triplet Loss with BERT Models*. *544 LNNS*, 520–532. https://doi.org/10.1007/978-3-031-16075-2_37

Pezeshkpour, P., Iso, H., Lake, T., Bhutani, N., & Hruschka, E. (2023). *Distilling Large Language Models using Skill-Occupation Graph Context for HR-Related Tasks*. http://arxiv.org/abs/2311.06383

Rezaeipourfarsangi, S., & Milios, E. E. (2023). *AI-powered Resume-Job matching: A document ranking approach using deep neural networks*. DocEng 2023 - Proceedings of the 2023 ACM Symposium on Document Engineering. https://doi.org/10.1145/3573128.3609347

Rojas-Galeano, S., Posada, J., & Ordoñez, E. (2022). A Bibliometric Perspective on AI Research for Job-Résumé Matching. *Scientific World Journal*, *2022*. https://doi.org/10.1155/2022/8002363

Shao, T., Song, C., Zheng, J., Cai, F., & Chen, H. (2023). Exploring Internal and External Interactions for Semi-Structured Multivariate Attributes in Job-Resume Matching. *International Journal of Intelligent Systems*, *2023*. https://doi.org/10.1155/2023/2994779

Shen, T., Geng, X., Tao, C., Xu, C., Long, G., Zhang, K., & Jiang, D. (2023). *UnifieR: A Unified Retriever for Large-Scale Retrieval*. 4787–4799. https://doi.org/10.1145/3580305.3599927

Shen, T., Long, G., Geng, X., Tao, C., Zhou, T., & Jiang, D. (2023). *Large Language Models are Strong Zero-Shot Retriever*. https://arxiv.org/pdf/2304.14233

Tanberk, S., Helli, S. S., Kesim, E., & Cavsak, S. N. (2023). *Resume Matching Framework via Ranking and Sorting Using NLP and Deep Learning*. 453–458. https://doi.org/10.1109/UBMK59864.2023.10286605

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). *LLaMA: Open and Efficient Foundation Language Models*. http://arxiv.org/abs/2302.13971

Vanetik, N., & Kogan, G. (2023). Job Vacancy Ranking with Sentence Embeddings, Keywords, and Named Entities. *Information (Switzerland)*, *14*(8). https://doi.org/10.3390/info14080468

Wang, L., Yang, N., & Wei, F. (2023). *Query2doc: Query Expansion with Large Language Models*. EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings. https://doi.org/10.18653/v1/2023.emnlp-main.585

Xiong, H., Bian, J., Li, Y., Li, X., Du, M., Wang, S., Yin, D., & Helal, S. (2024). *When Search Engine Services meet Large Language Models: Visions and Challenges*. http://arxiv.org/abs/2407.00128

Yao, K., Zhang, J., Qin, C., Wang, P., Zhu, H., & Xiong, H. (2022). *Knowledge Enhanced Person-Job Fit for Talent Recruitment*. *2022-May*, 3467–3480. https://doi.org/10.1109/ICDE53745.2022.00325

Zhang, M., Jensen, K. N., Sonniks, S. D., & Plank, B. (2022). *SkillSpan: Hard and Soft Skill Extraction from English Job Postings*. http://arxiv.org/abs/2204.12811

Zhang, Y., Liu, B., & Qian, J. (2023). FedPJF: federated contrastive learning for privacy-

preserving person-job fit. *Applied Intelligence*, *53*(22). https://doi.org/10.1007/s10489-023-04775-2

Zheng, Z., Qiu, Z., Hu, X., Wu, L., Zhu, H., & Xiong, H. (2023). *Generative Job Recommendations with Large Language Model*. https://arxiv.org/pdf/2307.02157

Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Chen, H., Dou, Z., & Wen, J.-R. (2023). *Large Language Models for Information Retrieval: A Survey*. http://arxiv.org/abs/2308.07107

Zhuang, S., Ma, X., Koopman, B., Lin, J., & Zuccon, G. (2024). *PromptReps: Prompting Large Language Models to Generate Dense and Sparse Representations for Zero-Shot Document Retrieval*. https://arxiv.org/pdf/2404.18424

Ziems, N., Yu, W., Zhang, Z., & Jiang, M. (2023). *Large Language Models are Built-in Autoregressive Search Engines*. Proceedings of the Annual Meeting of the Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.167