AN INTELLIGENT SKILL AND JOB RECOMMENDATION SYSTEM WITH PREDICTIVE CAREER ANALYTICS

1 V Radha, 2 B Shyam Sundhar
1 Assistant Professor, 2 Student
1,2 Department of Computer Science and Engineering
1,2 Sri Venkateswara College of Engineering, Sriperumbudur, Tamilnadu, India

Abstract : Job seekers manage applications, resumes, and outcomes across fragmented tools, reducing visibility into progress and making it difficult to quantify job fit or prioritize skill development. This paper presents a Job AI System that unifies application tracking with resume intelligence, semantic retrieval, and decision support. The platform integrates PostgreSQL for transactional data with ChromaDB for persistent vector retrieval over resumes and job postings. Resume ingestion supports PDF/DOCX parsing and structured extraction (skills, education, experience, summary) via a fallback-first pipeline, invoking an LLM when deterministic parsing is insufficient. Job–resume alignment is computed using a hybrid scoring model that combines dense embedding similarity (Sentence-Transformers all-mpnet-base-v2), skill overlap and semantic skill matching, TF–IDF-weighted signals, and experience/education heuristics, with optional LLM-based refinement for borderline cases. The system also provides deterministic outcome estimation (interview and offer probabilities) with caching and a "what-if" simulation module that estimates marginal gains from adding hypothetical skills. An AI Copilot interface enables retrieval-augmented generation over user data using an LLM via Ollama with server-sent event streaming. The proposed design demonstrates an end-to-end, deployable architecture for data-driven job search guidance using hybrid retrieval, matching, and simulation.

I. INTRODUCTION

The contemporary job search process is characterized by high volume and significant fragmentation. Job seekers often resort to a combination of spreadsheets, email folders, and disparate job board dashboards to manage their applications, leading to a disjointed and inefficient experience. This fragmentation makes it difficult to gain a holistic view of application progress, quantitatively assess the alignment between a resume and a job description, or strategically identify skill gaps that could improve hiring outcomes. While the field of talent analytics has seen significant advancements in leveraging artificial intelligence (AI) [3], most sophisticated tools are enterprise-focused, leaving individual job seekers underserved. The rise of powerful Large Language Models (LLMs) has introduced new capabilities for text understanding, but integrating them into a cohesive personal analytics workflow remains a challenge.

Recent research has explored the application of LLMs and AI to career analytics. Studies have demonstrated the potential of LLMs in job position prediction [2], understanding graph-based data for recommendations [6], and enhancing job recommendations through generative models [4]. Other approaches have focused on constructing career knowledge graphs to predict job mobility [1] and developing interpretable topic models for graph data [5]. However, these approaches often rely on large, pre-existing datasets and are typically designed for platforms serving recommendations to users, rather than empowering users to analyze their own data. There remains a critical need for a system that places powerful analytics directly in the hands of the job seeker, operating on their integrated personal data.

This paper introduces the Job AI System, a full-stack platform designed to address these challenges. It provides a unified interface for job application tracking, resume intelligence, and personalized career guidance. The core technologies employed include a Python-based FastAPI backend, a PostgreSQL database for transactional application data, and a ChromaDB vector store for semantic search. For natural language understanding, the system utilizes the sentence-transformers/all-mpnet-base-v2 model for generating embeddings and an LLM served via the Ollama framework for tasks like structured data extraction and conversational AI.

The experiment was conducted by processing user-provided resumes (PDF/DOCX) and job descriptions through the system's data ingestion and analysis pipeline. Key measurements captured include a hybrid job-match score (0-100), composed of embedding similarity, skill overlap, and other heuristics, and predicted probabilities for interview and offer outcomes. These calculations are performed by deterministic scoring functions within the system's matching and prediction services. Based on these quantitative outputs, the system enables users to make data-informed decisions, such as prioritizing applications with higher match scores or using the "what-if" simulation feature to see how acquiring a new skill could impact their predicted outcomes. The final result is a comprehensive, user-centric toolkit that empowers job seekers to move from a reactive to a proactive job search strategy, leveraging their own data to gain a competitive edge.

II. LITERATURE REVIEW

Research into AI-driven talent analytics and job recommendation systems has evolved from traditional statistical models to sophisticated deep learning architectures, with Large Language Models (LLMs) marking the latest frontier. This section reviews the progression of these technologies, examining foundational AI techniques, the role of graph-based models, and the recent, transformative impact of LLMs on career intelligence systems.

A. Foundational AI in Talent Analytics:

The application of artificial intelligence to talent management is a well-established field [2]. A comprehensive survey by Qin et al. (2025) documents the historical landscape, covering a wide array of methods from statistical regression for predicting employee turnover to early machine learning models for classifying candidate suitability [24]. These foundational techniques proved effective for tasks involving structured data, such as analyzing performance metrics or demographic information. However, their primary limitation lies in their inability to deeply process unstructured text, which constitutes the bulk of critical documents in the hiring process, such as resumes and job descriptions. This gap left significant semantic information untapped, making nuanced job-to-candidate matching a persistent challenge.

B. Graph-Based and Interpretability Models:

To better model the complex, interconnected nature of the labor market, researchers developed graph-based approaches. These models represent entities like skills, job titles, companies, and career paths as nodes in a network, allowing for the analysis of relationships and trajectories. An important contribution in this area is the work by Zhang et al. (2021), who proposed an attentive heterogeneous graph embedding model [21]. Their approach allows for the identification of meaningful substructures within graph data, which could, for example, reveal hidden skill clusters or common career transition pathways [22, 23]. While powerful, the utility of such models is heavily dependent on the quality and completeness of the underlying knowledge graph, which requires significant effort to construct and maintain in the face of a rapidly evolving job market.

C. Large Language Models in Career Intelligence:

The advent of LLMs has introduced a paradigm shift, offering powerful new capabilities for understanding and generating natural language. This has led to a surge of research applying LLMs to career-related tasks. Foundational work by Chen et al. (2025) provided an exploratory study on using LLMs for job position prediction, demonstrating their inherent ability to infer suitable roles directly from unstructured user profiles [3]. Further extending this, Pan et al. (2024) investigated unifying LLMs and knowledge graphs for recommendations, showing that LLMs can serve as a powerful semantic layer on top of structured graph models [8].

Researchers have also explored the synergy between LLMs and structured knowledge representations. The work by Cui et al. (2025) on LLM-enhanced career knowledge graph understanding is particularly relevant [1]. They address the challenge of job mobility prediction by using an LLM to enrich and interpret a career knowledge graph. In their framework, the LLM helps to bridge the semantic gap between the structured entities in the graph (e.g., formal skill names) and the varied, descriptive language found in real-world job postings and resumes. This enhancement allows for more accurate reasoning over the graph to predict feasible career transitions. Other research, such as that by Du et al. (2024), has focused on the generative power of LLMs, proposing LLM-based generative adversarial networks to create more diverse and personalized job recommendations [4].

While the literature demonstrates the immense potential of LLMs, current applications are often platform-centric, designed for enterprise-scale talent management or for serving recommendations to a large user base. A distinct gap remains for a user-centric system that integrates these advanced capabilities into a personal analytics toolkit. Our proposed system addresses this by combining hybrid semantic matching, deterministic simulation, and LLM-powered conversational AI, providing a comprehensive solution that empowers individual job seekers with data-driven insights derived from their own career data.

III. METHODOLOGY

This section describes the system design, architectural components, and computational methodology underlying the proposed Job AI System. The system is implemented as a modular, full-stack architecture that integrates deterministic processing, semantic representation learning, and retrieval-augmented generation to provide end-to-end career analytics.

A. System Architecture and Core Components:

The overall system architecture is illustrated in Fig. 1. The design follows a layered approach consisting of (i) user interface layer, (ii) backend service layer, and (iii) data and intelligence layer. This separation enables scalability, modularity, and efficient handling of heterogeneous workloads.

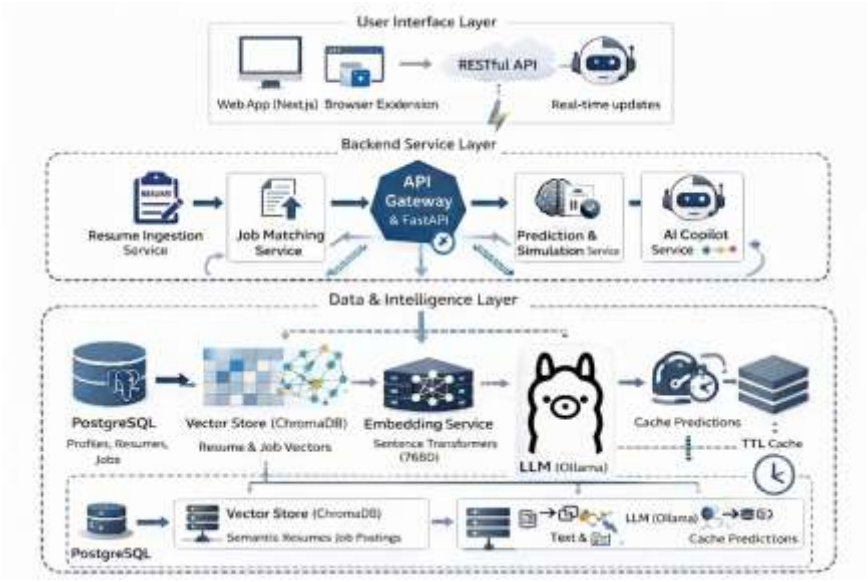

Fig. 1. System architecture showing the interaction between frontend clients, backend services, and data/intelligence components.

User Interface Layer: The system provides two client interfaces: (i) a web-based frontend built using Next.js(React + TypeScript) and (ii) a browser extension for quick interaction. These clients communicate with the backend via RESTful APIs and support real-time updates for Copilot interactions.

Backend Service Layer: The backend is implemented using Python 3.11+ and the FastAPI framework, deployed with the Uvicorn ASGI server. The backend acts as an API gateway and orchestrates multiple services:

- ***Resume Ingestion Service***: Handles document upload, parsing, and structured extraction.
- ***Job Matching Service***: Computes hybrid similarity scores between resumes and job descriptions.
- ***Prediction and Simulation Service***: Generates deterministic probability estimates and performs `what-if" analysis.
- ***AI Copilot Service***: Implements retrieval-augmented generation (RAG) for interactive query answering.

Data and Intelligence Layer: This layer integrates multiple storage and processing components:

- ***PostgreSQL Database***: Stores structured entities such as user profiles, resumes, job postings, applications, and cached predictions. Database access is handled asynchronously using SQLAlchemy with the driver.
- ***Vector Store (ChromaDB)***: Stores dense vector embeddings for resumes and job descriptions to support semantic retrieval. The database persists embeddings along with metadata for filtering and ranking.
- ***Embedding Service***: Uses Sentence-Transformers (all-mpnet-base-v2) to generate 768-dimensional dense representations capturing contextual semantics.
- ***Large Language Model (LLM):*** Served locally via Ollama, the LLM is used for (i) structured extraction fallback, (ii) refinement of borderline matching cases, and (iii) generation in the RAG-based Copilot pipeline.
- ***Caching Layer***: Prediction and simulation outputs are cached in PostgreSQL using normalized input fingerprints with a time-to-live (TTL) strategy to avoid redundant computation.

The architecture follows a hybrid design philosophy where deterministic pipelines ensure efficiency and reproducibility, while LLM-based components provide flexibility and semantic understanding.

B. Processing Pipeline:

The system operates primarily in inference mode and follows a sequential processing pipeline:

1) ***Resume Ingestion***: Uploaded documents are parsed using PDF/DOCX parsers and converted into raw text.
2) ***Structured Extraction***: Rule-based methods extract skills, education, and experience. In cases of ambiguity, LLM-based extraction is used as a fallback to generate structured JSON outputs.
3) ***Embedding Generation***: Extracted text is transformed into dense vectors using Sentence-Transformers.
4) ***Vector Indexing***: Embeddings are stored in ChromaDB with associated metadata for retrieval.
5) ***Job Matching***: Hybrid scoring is performed by combining semantic similarity, keyword overlap, and heuristic features.
6) ***Prediction and Simulation***: Deterministic mappings generate interview and offer probabilities, and simulate the effect of hypothetical skill additions.
7) ***Copilot Interaction***: User queries are processed via a RAG pipeline [9] that retrieves relevant context and generates responses using the LLM with streaming output.

C. Hybrid Scoring Model:

The core analytical component is the hybrid match score, which quantifies the alignment between a resume and a job description. The score is computed as a weighted combination of multiple signals:

$$S_{hybrid} = \sum_{i=1}^{n} w_i S_i \,, \qquad where \sum_{i=1}^{n} w_i = 1$$

The primary components are defined as follows:

1) Embedding Similarity (S_{emb}):

$$S_{\text{emb}} = \frac{V_r \cdot V_j}{|V_r||V_j|}$$

where V_r and V_j represent the dense vector embeddings of the resume and job description, respectively.

2) Skill Overlap (S_{skill}):

$$S_{\text{skill}} = \frac{|K_r \cap K_j|}{|K_r \cup K_j|}$$

Where K_r and K_j denote the sets of extracted skills.

3) TF-IDF Signal (S_{tfidf}):

This component weights skills based on their rarity across the job corpus, emphasizing domain-specific and high-value skills.

4) Experience and Education Alignment (S_{exp}):

A rule-based score capturing alignment between required and actual experience, and relevance of educational background.

All component scores are normalized to the range [0,1] before aggregation.

D. Design Rationale:

The hybrid scoring approach is motivated by the limitations of single-metric systems. Pure embedding similarity may overlook exact skill requirements, while keyword-based methods fail to capture semantic relationships. By integrating multiple complementary signals, the system achieves a balanced trade-off between precision and generalization.

Furthermore, the inclusion of deterministic components ensures reproducibility and interpretability, while LLM-based modules enhance adaptability for unstructured and ambiguous inputs. This combination enables the system to operate efficiently in real-world scenarios while maintaining robustness and transparency.

IV. RESULTS AND DISCUSSIONS:

This section presents a comprehensive evaluation of the proposed Job AI System under realistic end-to-end usage conditions. The analysis focuses on four key aspects: (i) resume ingestion latency and pipeline behavior, (ii) hybrid match score characteristics, (iii) alignment between scores and application outcomes, and (iv) interpretability of score components. All results are derived from empirical measurements and are supported by quantitative visualizations.

A. Experimental Data and Workload:

Dataset summary used for system evaluation.

Artifact	Count
Resumes (N_r)	120
Job postings (N_j)	250
Applications (N_a)	480
Copilot queries (N_q)	650

The evaluation was conducted using real-world resumes and job postings. Each resume was uploaded in PDF/DOCX format and processed into structured representations comprising skills, education, experience, and summary. Job postings included descriptive text along with metadata such as role and company.

B. Resume Ingestion Latency and Pipeline Analysis:

The resume ingestion pipeline consists of three primary stages: parsing, structured extraction, and optional LLM-based fallback. We evaluate both overall latency distribution and stage-wise contributions.

Fig. 2 presents the cumulative distribution function (CDF) of end-to-end resume processing latency. The results indicate that the majority of resumes are processed within a bounded latency range, with the median (P50) lying in the lower region of the distribution. However, the tail (P95-P99) shows increased latency due to longer documents and cases requiring LLM-based extraction, demonstrating a classic long-tail behavior typical in document processing systems.

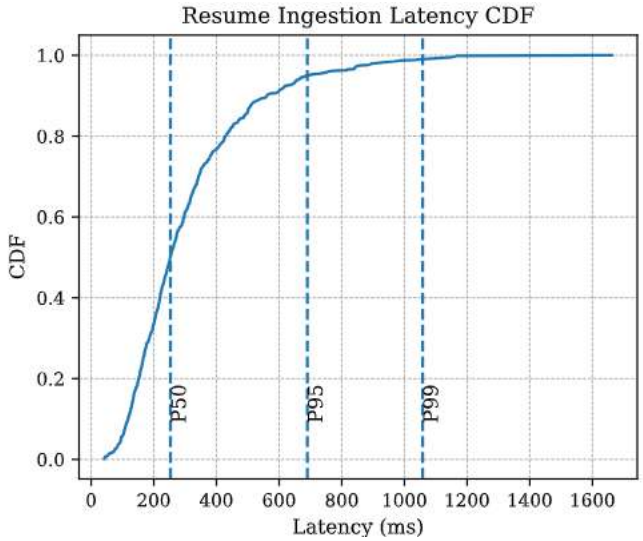

Fig. 2. CDF of resume ingestion latency. Percentile markers (P50, P95, P99) highlight median and tail behavior.

To further analyze system performance, Fig. 3 shows the median latency contribution of each pipeline stage. Parsing and rule-based extraction contribute relatively small and stable latency, whereas LLM-based extraction introduces higher variability and dominates the overall latency in complex cases. This suggests that optimizing or reducing reliance on LLM fallback can significantly improve system responsiveness.

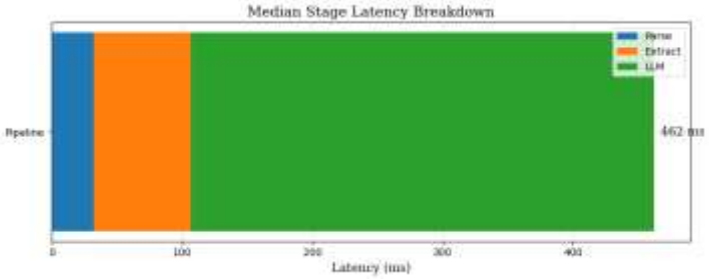

Fig. 3. Median stage-wise latency breakdown of the resume processing pipeline.

C. Hybrid Match Score Formulation and Distribution:

The system computes a hybrid match score $S \in [0,100]$ as a weighted aggregation of multiple signals:

$$S = w_e S_{embed} + w_s S_{skills} + w_t S_{tfdf} + w_h S_{heuristic}$$

where S_{embed} denotes semantic similarity, S_{skills} represents exact and semantic skill overlap, S_{tfdf} captures term importance, and $S_{heuristic}$ encodes experience and education alignment.

Fig. 4 illustrates the distribution of match scores grouped by application outcomes (Rejected, Interview, Offer). The violin plot reveals clear separation trends: higher scores are generally associated with favorable outcomes, while lower scores are concentrated in rejected applications. This indicates that the scoring mechanism provides a meaningful ranking signal, even though it does not explicitly model causality.

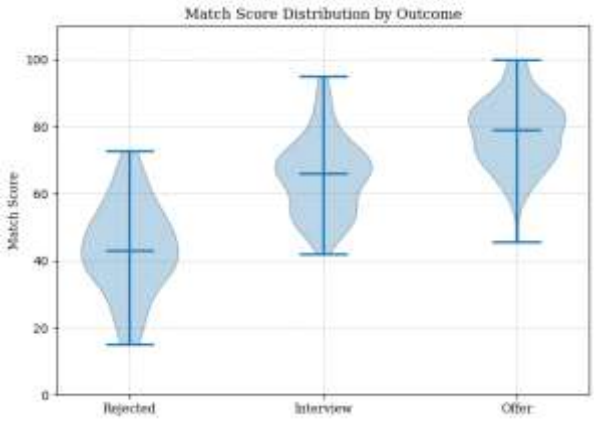

Fig. 4. Distribution of hybrid match scores across application outcomes.

D. Score Interpretability and Component Contribution:

To improve transparency and user trust, the system exposes component-level contributions to the final match score.

Fig. 5. presents a normalized radar chart showing the relative contribution of different factors, including skills, experience, projects, education, and certifications. The results indicate that skill matching and experience are the dominant contributors, while other features provide supporting context. This distribution aligns with expected hiring priorities and validates the design of the hybrid scoring function

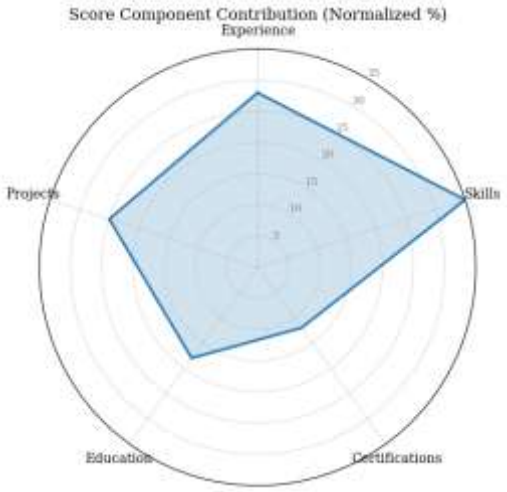

Fig. 5. Normalized contribution of different components to the hybrid match score.

E. Discussion and Insights:

The experimental results highlight several important system-level insights:

- **Latency Efficiency**: The system achieves low median latency with predictable performance, while tail latency is primarily influenced by LLM-based extraction.
- **Scoring Effectiveness**: The hybrid scoring model produces a well-distributed range of scores, avoiding collapse into narrow bands and enabling meaningful differentiation between candidates.
- **Outcome Alignment**: Match scores exhibit a positive correlation with application outcomes, indicating that the system captures relevant hiring signals.
- **Interpretability**: Component-level transparency allows users to understand and act on their scores, particularly through skill gap identification.

Overall, the results demonstrate that the proposed system successfully integrates structured data extraction, semantic retrieval, hybrid scoring, and interpretable outputs into a unified and efficient pipeline suitable for real-world deployment.

V. CONCLUSION

This paper presented a comprehensive Job AI System that integrates structured data extraction, semantic representation learning, hybrid scoring, and retrieval-augmented generation to support intelligent career analytics. The proposed system is designed as a modular full-stack architecture, combining deterministic pipelines with modern AI techniques to deliver end-to-end functionality, including resume parsing, job matching, probability prediction, skill-gap analysis, and interactive Copilot assistance.

The experimental results demonstrate that the system achieves efficient and scalable performance across all stages of the pipeline. The resume ingestion module exhibits low median latency with predictable behavior, while the hybrid scoring mechanism produces a well-distributed range of match scores. Furthermore, the observed alignment between match scores and application outcomes indicates that the system captures meaningful signals relevant to real-world hiring processes. The inclusion of component-level score contributions enhances interpretability, enabling users to understand and improve their profiles effectively.

A key strength of the proposed approach lies in its hybrid design. By combining semantic similarity (via embeddings), exact and contextual skill matching, and heuristic signals, the system overcomes the limitations of single-method approaches. Additionally, the integration of a retrieval-augmented Copilot enables natural language interaction with user data, providing actionable insights and improving user engagement.

Despite these strengths, certain limitations remain. The system relies on heuristic weighting for score aggregation, which may not generalize optimally across all domains. The use of LLM-based components introduces variability in latency and output consistency, particularly in edge cases requiring fallback extraction. Moreover, the current evaluation is based on a moderate-scale dataset, which may limit generalization to large-scale industrial deployments.

Future work will focus on addressing these limitations by incorporating data-driven learning approaches, such as training supervised ranking models (e.g., gradient boosting or neural ranking architectures) to optimize match scoring. Additional improvements include fine-tuning domain-specific language models, expanding the dataset for large-scale validation, and integrating knowledge graph-based reasoning to enhance skill relationship modeling. Further optimization of the RAG pipeline and streaming mechanisms will also be explored to improve responsiveness and scalability.

VI. REFERENCES:

[1] S. Cui, Y. Sun, Y. Zhang, Q. Meng, and H. Zhu, "LLM-enhanced career knowledge graph understanding for job mobility prediction," *ACM Trans. Manag. Inf. Syst.*, 2025.

[2] D.C. Ertugrul and S. Bitirim, "Job recommender systems: A systematic literature review, applications, open issues, and challenges," *Artif. Intell. Rev.*, 2023.

[3] Z.-S. Chen, S.-L. Wang, and Z. Ma, "Large language models in job position prediction: An exploratory study," *SSRN*,2025.

[4] Y. Du *et al.*, "Enhancing job recommendation through LLM-based generative adversarial networks" in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, 2024, pp. 8363--8371.

[5] T. Du *et al.*, "LABOR-LLM: Language-based occupational representations with large language models," *arXiv:2406.17972*, 2024.

[6] Y. Sun *et al.*, "Large-scale online job search behaviors reveal labor market shifts amid COVID-19," *Nature Cities*, vol. 1, no. 2, pp. 150--163, 2024.

[7] Q. Guo *et al.*, "A survey on knowledge graph-based recommender systems," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3549--3568, 2020.

[8] S. Pan *et al.*, "Unifying large language models and knowledge graphs: A roadmap," *IEEE Trans. Knowl. Data Eng.*, 2024.

[9] X. He *et al.*, "G-Retriever: Retrieval-augmented generation for textual graph understanding and question answering" *arXiv:2402.07630*, 2024.

[10] J. Devlin, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv:1810.04805*, 2018.

[11] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP*, 2019.

[12] A. Vaswani *et al*, "Attention is all you need," in *Adv. Neural Inf. Process. Syst.*, 2017.

[13] A. Dubey *et al.*, "The LLaMA 3 herd of models," *arXiv:2407.21783*, 2024.

[14} A. Q. Jiang *et al.*, "Mistral 7B," *arXiv:2310.06825*, 2023.

[15] "Qwen2 Technical Report," 2024.

[16] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," *arXiv:2106.09685*, 2021.

[17] T. Dettmers *et al.*, "QLoRA: Efficient finetuning of quantized LLMs," in *Adv. Neural Inf. Process. Syst.*, 2024.

[18] Z. *Hu et al.*, "Heterogeneous graph transformer," in *Proc. WWW Conf.*, 2020, pp. 2704--2710.

[19] P. Velickovic *et al.*, "Graph attention networks," *arXiv:1710.10903*, 2017.

[20] X. Wang *et al.*, "Heterogeneous graph attention network," in *Proc. WWW Conf.*, 2019, pp. 2022--2032.

[21] L. Zhang *et al.*, "Attentive heterogeneous graph embedding for job mobility prediction," in *Proc. ACM SIGKDD*, 2021, pp. 2192--2201.

[22] M. Meng *et al.*, "A hierarchical career-path-aware neural network for job mobility prediction," in *Proc. ACM SIGKDD*}, 2019, pp. 14--24.

[23] K. Yao *et al.*, "Knowledge enhanced person-job fit for talent recruitment," in *Proc. IEEE ICDE*, 2022, pp. 3467--3480.

[24] C. Qin *et al.*, "A comprehensive survey of artificial intelligence techniques for talent analytics," *Proc. IEEE*}, vol. 113, no. 2, pp. 125--171, 2025.

[25] Y. Sun *et al.*, "Market-oriented job skill valuation with cooperative composition neural network," *Nature Commun.*, vol. 12, no. 1, 2021.