RESUME PARSER USING KNOWLEDGE GRAPH AND RAG

Catherene Joshi,
Student,
Department of Computer Science
Shri JJT University, Rajasthan.

Dr Kelapati,
Assistant Professor,
Department of Computer Science,
Shri JJT University,
Rajasthan.

Abstract— *The increasing demand for efficient talent acquisition has driven the adoption of automated tools for resume parsing, a critical component in recruitment workflows. This dissertation explores the integration of Large Language Models (LLMs) and Knowledge Graphs to design an advanced resume parser capable of extracting, structuring, and contextualizing information with unprecedented accuracy. Leveraging the contextual understanding and natural language processing capabilities of LLMs, the proposed system transcends traditional keyword-based parsing by interpreting nuanced details such as career progression, domain expertise, and skill associations. Complementing this, Knowledge Graphs provide a structured representation of relationships between entities, enabling the system to infer and validate connections, such as linking ambiguous skill terminologies to standardized categories or identifying hierarchical relationships within job roles. The architecture combines LLMs for semantic analysis with Knowledge Graphs for relational reasoning, ensuring scalability, adaptability to diverse resume formats, and enhanced interpretability. Real-world applicability is validated through case studies in recruitment pipelines, demonstrating the system's ability to streamline candidate selection and reduce manual intervention.*

Index Terms— Extraction, Knowledge graphs, Large Language Models, Retrieval Augmented Generation

I. INTRODUCTION

In the modern landscape of employment, where competition among candidates is fiercer than ever, the recruitment process has become both a critical and increasingly complex task for organizations. Hiring the right talent not only determines the efficiency and productivity of a business but also significantly impacts its culture and long-term growth. As companies receive an overwhelming number of job applications for every opening, the task of sifting through resumes and identifying the most suitable candidates has grown more demanding and intricate. This exponential surge in application volume, coupled with the diverse and often ambiguous nature of candidate backgrounds, has rendered manual resume screening methods not only inefficient but also highly susceptible to human error, fatigue, and unconscious biases.

Traditionally, recruiters have relied on manual review or simple rule-based software to filter resumes based on keywords, job titles, or educational qualifications. However, these approaches often lack the sophistication required to grasp the subtle context or the implicit meaning embedded within a candidate's resume. They fail to consistently interpret non-standard formats, unique career trajectories, or evolving job roles. Consequently, highly qualified candidates may be overlooked due to unconventional presentation, while less suitable applicants

might pass initial filters simply because they use the "right" keywords.

The proposed hybrid model offers a transformative approach to resume parsing, combining the deep semantic understanding of LLMs, the structured reasoning of Knowledge Graphs, and the flexible, context-driven processing capabilities of RAG. Together, these technologies establish a high-performance, AI-driven solution tailored to meet the growing complexity of recruitment, ensuring smarter, fairer, and faster hiring practices.

II. LITERATURE SURVEY

V. Gulati et al [4], in their paper presents a Resume Analyzer that leverages Natural Language Processing (NLP) techniques to enhance the efficiency of resume screening and job matching. They have identified the key gaps in existing research on automated resume screening, such as the lack of discussion on ethical issues, insufficient diversity across career fields, and the absence of a dialogue on the limitations of current methodologies. The paper emphasizes the importance of addressing these gaps to create more inclusive and effective hiring processes.

E. Kaygin [1] presents a comprehensive analysis of the effectiveness of resume parsing and matching algorithms, particularly focusing on the integration of Large Language Models (LLM), Natural Language Processing (NLP), and Optical Character Recognition (OCR) technologies. It highlights the limitations of traditional algorithms, which often struggle with complex resume formats and fail to accurately extract relevant information, leading to inefficiencies in recruitment processes. The study employs a mixed-methods approach, analyzing datasets from various industries to evaluate the performance of advanced algorithms compared to traditional methods, ultimately aiming to enhance the accuracy of candidate evaluations in Applicant Tracking Systems (ATS). The findings indicate that the combined use of LLM and OCR significantly improves the accuracy of data extraction from resumes, effectively addressing the semantic gaps and structural complexities that traditional algorithms encounter. The research underscores the potential of these advanced technologies to streamline the recruitment process, reduce manual labor, and ensure that qualified candidates are not overlooked due to formatting discrepancies. The paper concludes by suggesting directions for future research to further explore the evolving landscape of recruitment technologies and their implications for talent acquisition.

The task of translating user questions into SQL queries for databases, known as text-to-SQL, is a vital area in natural language processing. The advent of large language models (LLMs) has significantly transformed this field, presenting both new opportunities and challenges. Z. Hong et al [3], in their survey aims to provide a comprehensive overview of the evolution of text-to-SQL systems, focusing on the impact of LLMs on their development and performance.

In the study, M Würsch et al [5], investigates the performance of various entity extraction models, particularly focusing on large language models (LLMs)

in the context of cybersecurity literature. Using a dataset of arXiv preprints, the research identifies that while LLMs exhibit similar performance due to their underlying architectures, they struggle to extract relevant knowledge entities specific to cybersecurity. In contrast, non-LLM-based extractors, such as Yake and spaCy, demonstrate more effective results, emphasizing the need for careful selection of models in bibliometric tasks.

Additionally, the research highlights the significant impact of embedding choices on the effectiveness of entity extraction. The analysis reveals that cosine similarity, used to assess the relatedness of extracted entities, is highly dependent on the embedding method employed. This finding underscores the challenges in automated processing of extracted terms and suggests that attention must be given to the choice of embeddings in entity extraction and analysis pipelines.

Sarmah, B., et al. [6] in their paper introduces HybridRAG, a novel approach combining Knowledge Graphs (KGs) and Vector Retrieval Augmented Generation (VectorRAG) to improve information extraction from financial documents. The methodology employs a two-tiered LLM chain for knowledge extraction, resulting in enhanced retrieval accuracy, answer relevancy, and context recall. The proposed methodology enhances information extraction by utilizing a two-tiered LLM chain that first generates abstract representations of document chunks, preserving essential information and relationships. It employs advanced prompt engineering for structured output, resulting in rich, multidimensional triplet representations that facilitate nuanced analysis. Additionally, the integration of entity disambiguation techniques improves the quality and utility of the extracted triplets, ensuring coherent knowledge graphs.

Chengguang Gan [7] et al in their paper explains the automated resume screening process that begins with the preparation of resumes, which are formatted with identifiers, grades, and summaries to simulate the selection of final candidates. It is divided into three stages: classification, where resume content is identified and categorized; grading and summarization, where scores are assigned based on relevant skills and key information is summarized; and decision making, where top candidates are selected based on their grades with justifications provided. Additionally, the framework is customizable and adaptable to various industries by modifying keywords and criteria, focusing on skills and work experience to minimize educational bias. The proposed resume screening framework employs a systematic approach that integrates Large Language Models (LLMs) for efficient resume evaluation. Initially, a dataset of resumes is prepared, focusing on key categories such as personal information, experience, education, and skills. The LLM, specifically the fine-tuned LLaMA2 model, is utilized to classify, grade, and summarize the resumes, with the process being divided into three stages: classification, grading & summarization, and decision-making. Human evaluators, trained to standardize the grading process, manually assess a subset of resumes to validate the LLM's performance against a gold standard established by GPT-4 annotations. This methodology not only

streamlines the screening process but also allows for adaptability across various sectors by modifying evaluation criteria.

III. PROBLEM STATEMENT

To enhance the efficiency and accuracy of information extraction from financial documents by integrating Knowledge Graphs (KGs) and Vector Retrieval Augmented Generation (VectorRAG) to improve contextual understanding and retrieval relevancy for applications such as resume parser.

IV. METHODOLOGY

The architecture of the proposed resume parser system is meticulously designed by leveraging state-of-the-art solutions. The system is optimized to handle large amounts of data efficiently, while ensuring an intuitive and seamless experience for end-users.

The first stage of the system's operation involves the critical task of loading large local language models (LLMs), specifically Meta 3.1 and mxbai-embed-large, using the Ollama framework. The selection of Meta 3.1 and mxbai-embed-large is based on their complementary capabilities, each offering a distinct set of strengths that are crucial to the system's overall performance and efficiency.

Meta 3.1, one of the primary models in use, is especially proficient in understanding language at a deep level and extracting relevant entities from the text. This process includes recognizing various entities such as names, dates, locations, and other significant terms that are often necessary for more advanced analysis or decision-making processes. Its ability to break down complex sentences and phrases ensures that even convoluted or nuanced statements are comprehended and interpreted accurately.

The local processing setup offers significant benefit: enhanced control over data privacy and security. By running these models on local machines, sensitive information does not need to be transmitted to external servers, which can be a major concern in many industries, especially those that handle private or regulated data.

The extraction process within llama3.1 is carried out using carefully crafted prompting techniques designed to ensure that the model captures all the key entities in a resume. For example, the model doesn't just extract technologies like VAPT (Vulnerability Assessment and Penetration Testing), C++, scikit-learn, and pandas but also takes into account contextual elements such as the duration of the candidate's experience with each specific technology. If a candidate lists pandas as a skill, the model doesn't just note its presence. Instead, it also aggregates the candidate's experience with pandas across various roles or organizations. For instance, if a candidate has worked with pandas for three years in one role and another two years in a different organization, the system intelligently combines these timelines to give a total of five years of experience with pandas. This aggregation of experience provides a more accurate and comprehensive picture of the candidate's expertise, offering recruiters a detailed understanding of their technical background.

This level of granularity in extracting and processing data is incredibly valuable, as it

allows recruiters to filter candidates with a much higher degree of precision. For example, rather than simply looking for candidates who list pandas as a skill, recruiters can now filter candidates based on a specific number of years of experience with pandas. This means that recruiters can request results showing only those candidates who have, say, five or more years of experience with pandas, enabling them to zero in on the most qualified individuals for the job. In this way, the system's ability to capture nuanced details about a candidate's experience significantly enhances the accuracy of search results during the hiring process. This transformation of unstructured resume data into structured, actionable insights greatly enhances the ability of recruiters to make informed decisions.

After entity extraction, the processed resume content is divided into smaller, overlapping chunks. The processed chunks of data are then converted into a knowledge graph using LangChain's LLMGraphTransformer. By leveraging LangChain's LLMGraphTransformer, the system is able to translate the extracted information into a knowledge graph that represents entities and their relationships as triplets. These triplets consist of a subject, predicate, and object, and they provide a way of representing data in a structured manner that is easy to interpret and use for further processing. For example, if the system is asked to find all resumes that mention a specific skill, it can simply look for triplets where the skill is the object, making the retrieval process quick and efficient.

Once the graph structures are generated, they are systematically stored in the Neo4j database by utilizing the add_graph_documents function. This function plays a crucial role in ensuring that each newly created document, which contains relevant information, is smoothly incorporated into the existing knowledge graph. By using this method, the system ensures a seamless integration process. It guarantees that the information from new documents doesn't conflict with or duplicate data already present in the graph, maintaining the integrity and accuracy of the entire knowledge base. Through this, the knowledge graph becomes a dynamic and constantly evolving structure, continuously updated as new documents are added. This dynamic feature ensures that the knowledge graph grows over time, enriching the repository with fresh and relevant data, without losing consistency or coherence.

Moreover, this approach prevents any redundancy, making sure that there is no unnecessary repetition of information within the system. Every piece of information is checked, and if it has already been stored, it is not re-entered, thus avoiding duplication. This helps in keeping the graph lean and efficient, enabling smooth operations when dealing with large volumes of data. The ongoing process of adding new information while preventing redundant entries allows the graph to stay consistent and up to date with the latest developments, without becoming cluttered or difficult to manage. The system defines various parameters, such as node_label, text_node_properties, and embedding_node_property, to effectively organize the graph nodes, ensuring that information is stored in an optimal way that

enhances both storage efficiency and retrieval performance.

To further improve the system's ability to extract meaningful insights, a full-text query function is implemented within Cypher. This function allows users to retrieve specific nodes and their associated relationships based on a set of defined queries. By querying the graph, users can access not only direct relationships but also neighboring connections, which provide a more comprehensive context for understanding the data. For example, a query using a candidate's first name may retrieve relevant resume details, even though the knowledge graph stores the full name. This capability makes it easier to obtain relevant information without needing to specify every detail, offering a more intuitive and flexible querying experience.

In addition to the full-text query function, LLMs are used to dynamically extract entities from user queries. This step involves cross-validating the extracted entities against the existing graph data to ensure that new entities enhance the graph's knowledge base without introducing redundancy. This dual-validation process adds an extra layer of reliability, ensuring that the information extracted from complex resumes is accurate and meaningful. By incorporating both dynamic entity extraction and graph-based cross-validation, the system increases its accuracy and relevance in identifying and understanding key information in resumes.

Finally, a specialized function is employed to fetch vector embeddings that are similar to the user's query. These embeddings are then combined with the results of the full-text graph query, creating a hybrid approach that improves retrieval precision. This method leverages both semantic similarity captured through the embeddings and structured relationships represented in the knowledge graph ensuring that the system can return the most relevant results based on both content and context.

The system also integrates vector-based indexing and embeddings. These embeddings are generated by mxbai-embed-large, a model that transforms resume data into high-dimensional vectors that capture the semantic meaning of the text. This allows the system to perform similarity-based searches, where queries can be matched not only to exact keywords but also to concepts with similar meanings. For example, a query for "data analysis" might retrieve resumes that mention "data science" or "data visualization," even if the exact phrasing differs. This vector-based indexing ensures that the system can retrieve results that are semantically relevant, even when the language used in the query is different from what is present in the resumes. The ability to perform such similarity-based searches greatly enhances the system's precision, making it more effective at matching candidates with the right skills and experience.

This solution streamlines the process of extracting meaningful insights from resumes, making it a valuable tool for recruiters, HR professionals, and hiring managers who need to process large volumes of resumes efficiently and accurately.

V. RESULTS

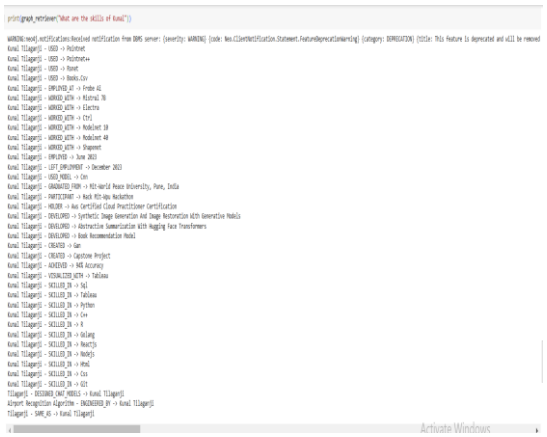

Fig 1: Output from graph retriever

Fig 2: Output from final chain with different query

This architecture enabled the system to answer complex, multi-faceted queries like: "Find candidates proficient in Python with over five years of experience in machine learning." In this scenario, the system correctly identified and retrieved relevant candidates, associating the skill (Python), the domain (machine learning), and the experience level (>5 years) across multiple data points.

VI. CONCLUSION

The research presented a comprehensive and successful exploration into the development and application of an advanced resume parser, demonstrating its efficacy through the integration of a hybrid model that brings together several advanced techniques. This model combines Knowledge Graphs (KG), full-text querying, and Retrieval-Augmented Generation (RAG), creating a powerful system capable of more accurate and context-aware information extraction. Throughout the research, the study meticulously examined the shortcomings of traditional approaches, such as the SpaCy framework, plain Large Language Models (LLMs), and Text-to-SQL methods, emphasizing the critical need for a more holistic solution that is capable of understanding and processing complex data in a more sophisticated manner. Traditional methods were shown to lack the ability to fully integrate and leverage diverse data sources effectively, often missing key contextual nuances that are essential for precision in information retrieval and extraction.

REFERENCES

[1] Kaygin, Esranur. (2023). Comparative Analysis of ML (Machine Learning) and LLM (Large Language Models) in Resume Parsing: A Paradigm Shift in Talent Acquisition. 10.13140/RG.2.2.22517.35047.

[2] Tran Q-B-H, Waheed AA, Chung S-T. Robust Text-to-Cypher Using Combination of BERT, GraphSAGE, and Transformer (CoBGT) Model. Applied Sciences. 2024; 14(17):7881. https://doi.org/10.3390/app14177881

[3] Z. Hong, Z. Yuan, Q. Zhang, H. Chen, J. Dong, F. Huang, X. Huang , "Next-Generation Database Interfaces: A Survey of LLM-based Text-to-SQL" arXiv:2406.08426

[4]V. Gulati, I. Gupta, "Resume Analyzer Using Natural Language Processing", http://ir.juit.ac.in:8080/jspui/jspui/handle/123456789/11318

[5] M. Würsch, A. Kucharavy, D. Percia-David and A. Mermoud, "LLM-Based Entity Extraction Is Not for Cybersecurity", https://ceur-ws.org/Vol-3451/paper5.pdf

[6] B. Sarmah, D. Mehta, B. Hall, R. Rao, S. Patel, S. Pasquali, "HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction", 2024, arXiv:2408.04948

[7] Chengguang Gan, Qinghao Zhang, Tatsunori Mori, "Application of LLM Agents in Recruitment: A Novel Framework for Automated Resume Screening", Journal of Information Processing Vol.32 881–893 (Oct. 2024), [DOI: 10.2197/ipsjjip.32.881]

[8] Radhakrishnan Venkatakrishnan, Emrah Tanyildizi, and M. Abdullah Canbaz. 2024. Semantic interlinking of Immigration Data using LLMs for Knowledge Graph Construction. In Companion Proceedings of the ACM Web Conference 2024 (WWW '24). Association for Computing Machinery, New York, NY, USA, 605–608. https://doi.org/10.1145/3589335.3651557

[9] How I created a Neo4j Search Engine with Generative AI, https://medium.com/@thcookieh/how-i-created-a-neo4j-search-engine-with-generative-ai-98b43cf8ec1e

[10] How Neo4j and Generative AI Work Together, https://www.linkedin.com/pulse/how-neo4j-generative-ai-work-together-sanjeewa-rathnayake-vaoyc/

[11] A Tale of LLMs and Graphs: The GenAI Graph Gathering, https://neo4j.com/developer-blog/genai-graph-gathering/

[12] Z. Xu, M. J. Cruz, M. Guevara, T. Wang, M. Deshpande, X. Wang, Z. Li "Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering", https://doi.org/10.48550/arXiv.2404.17723

[13] Polak, M.P., Morgan, D. Extracting accurate materials data from research papers with conversational language models and prompt engineering. Nat Commun 15, 1569 (2024). https://doi.org/10.1038/s41467-024-45914-8

[14] L. Schmidt, K. Hair, S. Graziosi, F. Campbell, C. Kapp, A. Khanteymoori, D. Craig, M. Engelbert, J. Thomas, "Exploring the use of a Large Language Model for data extraction in systematic reviews: a rapid feasibility study", Proceedings of the 3rd Workshop on Augmented Intelligence for Technology-Assisted Reviews Systems (ALTARS 2024), Glasgow. https://ceur-ws.org/Vol-3832/paper2.pdf

[15] A. Herandi, Y. Li, Z. Liu, X. Hu, X. Cai, "Skill-LLM: Repurposing General-Purpose LLMs for Skill Extraction", https://arxiv.org/html/2410.12052v1

[16] "Why Text-to-SQL Isn't Enough: The Case for Knowledge Graphs in Data Management", https://www.cloudgeometry.com/blog/why-text-to-sql-isnt-enough-the-case-for-knowledge-graphs-in-data-management

[17] "Leveraging SQL Knowledge Graphs for Accurate LLM SQL Query Generation", https://timbr.ai/blog/leveraging-sql-knowledge-graphs-for-accurate-llm-sql-query-generation/

[18] J. F. Sequeda, D. Allemang, B. Jacob, "A benchmark to understand the role of knowledge graphs on large language model's accuracy for question answering on enterprise SQL databases", https://arxiv.org/pdf/2311.07509

[19] A. Alcaraz, "Knowledge Graphs Drastically Improve Accuracy of Large Language Models on Complex Data: New Research Proves", https://iianalytics.com/community/blog/knowledge-graphs-improve-accuracy-of-large-language-models

[20] T. Nijhof, "Full-Text Search in 197M Chemical Names Graph Database", https://neo4j.com/blog/developer/full-text-search-chemical-names/

[21] H. Ishimwe, "Finding the Best Open-Source Embedding Model for RAG", https://www.timescale.com/blog/finding-the-best-open-source-embedding-model-for-rag

[22] M. Ozkaya, "Exploring Vector Embedding Models: OpenAI's and Ollama's Offerings", https://mehmetozkaya.medium.com/exploring-vector-embedding-models-openais-and-ollama-s-offerings-831799f4dc74

[23] A. Grattafiori et al., "The Llama 3 Herd of Models", https://doi.org/10.48550/arXiv.2407.21783

[24] M. Malec, "Open-Source LLMs vs Closed: Unbiased Guide for Innovative Companies", https://hatchworks.com/blog/gen-ai/open-source-vs-closed-llms-guide/

[25] F. Liu, Z. Kang, X. Han, "Optimizing RAG Techniques for Automotive Industry PDF Chatbots: A Case Study with Locally Deployed Ollama Models Optimizing RAG Techniques Based on Locally Deployed Ollama ModelsA Case Study with Locally Deployed Ollama Models", AIIIP 2024: 2024 3rd International Conference on Artificial Intelligence and Intelligent Information Processing, http://dx.doi.org/10.1145/3707292.3707358