2025 2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN) | 979-8-3315-3038-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/CICTN64563.2025.10932524

# Leveraging RAG for Effective Prompt Engineering in Job Portals

Farhan Haneef
*School of Computer Science and Engineering*
*Vellore Institute of Technology*
Vellore, Tamil Nadu
farhan.haneef2021@vitstudent.ac.in

Varalakshmi M
*Associate Proffessor, School of Computer Science and Engineering*
*Vellore Institute of Technology*
Vellore, Tamil Nadu
mvaralakshmi@vit.ac.in

Peer Mohamed P U
*Engineering Leader*
*CapitalOne*
USA
pupeer@gmail.com

***Abstract*—Traditional recruitment methods are time-consuming and heavily reliant on manual processes, leading to inefficiencies. This paper explores the integration of AI technologies into a job portal to streamline the hiring process, focusing on AI's ability to automatically parse and extract information from resumes and job descriptions to match candidates with relevant job opportunities. The platform leverages Large Language Models (LLMs) to extract key information, such as skills, education, and work experience, from both resumes and job descriptions. Fine-tuning through prompt engineering ensures accurate extraction, even when data fields are incomplete or missing. To further enhance matching accuracy, Retrieval-Augmented Generation (RAG) techniques are employed. These mechanisms retrieve relevant information from a structured skills database to provide context, which, combined with the generative capabilities of LLMs, enables more contextually accurate matches. Candidates receive personalized job recommendations based on the information extracted from their resumes, while employers can post job descriptions and use AI-driven tools to match candidates. This approach of using contextual prompts not only improves matching accuracy but also reduces computational time, eliminating the need for custom models tailored specifically to resumes or job descriptions.**

***Keywords—Retrieval-Augmented Generation (RAG), Large Language Models (LLM), Artificial Intelligence (AI), Job Portal, Prompt Engineering, TF-IDF.***

## I. INTRODUCTION

With the digitalization of recruitment, traditional methods like manually re-viewing resumes and matching candidates to job descriptions have become inefficient and time-consuming. The growing volume of job postings and candidates requires more automated solutions. Artificial Intelligence (AI) and Natural Language Processing (NLP) can automate the extraction of key information from resumes, improving candidate-job matching. Recent advancements in Large Language Models (LLMs) enable efficient parsing of complex resume data, extracting relevant details like skills, education, and experience. When combined with Retrieval-Augmented Generation (RAG) techniques, these models improve the accuracy of matching candidates to job requirements, enhancing both speed and efficiency in the recruitment process.

Despite advancements, many automated job portals still face challenges in processing resumes quickly and accurately, often suffering from delays, bias [1][2], the potential for algorithmic bias and discrimination[11]. These limitations prolong recruitment cycles and may lead to missed opportunities for candidates and employers [3]. Currently, most systems rely on fine-tuned or custom-trained models tailored to specific needs and overcoming these issues is crucial to improving the speed, accuracy and overall effectiveness of job matching. By leveraging advanced AI techniques such as LLMs and RAG, this paper aims to address these inefficiencies and offer a more responsive, user-friendly solution.

This paper presents a system that incorporates AI techniques to improve resume and job description parsing and candidate-job matching. It specifically addresses the shortcomings of existing systems that, despite automation, still struggle with delays and inaccuracies. The use of RAG techniques enhances the accuracy of the matching process by ensuring that extracted information is contextually relevant and accurate.

## II. RELATED WORK

Z.Hu in [4] investigates the robustness of Retrieval-Augmented Generation (RAG)-based Large Language Models (LLMs) against prompt perturbation at-tacks. They introduce Gradient Guided Prompt Perturbation (GGPP), a technique to manipulate RAG-based models by crafting misleading prefixes. The authors also propose two detection methods, Sentence Attention-based Probe (SATe) and Activation Probe (ACT), which leverage internal LLM states to identify perturbed prompts, enhancing robustness. Their extensive empirical analysis demonstrates the effectiveness of these methods in both manipulating and detecting prompt attacks, offering valuable insights for improving the trustworthiness of RAG-based systems.

Fan et al. [5] explores Retrieval-Augmented Large Language Models (RA-LLMs), which integrate information retrieval to enhance model knowledge. The paper examines RA-LLM components—retrieval, generation, and augmentation—and discusses their architectures, training strategies, and applications in tasks such as question answering, chatbots, and recommendations. It also outlines future research directions, focusing on developing trustworthy, multilingual, and multi-modal RA-LLMs and improving external knowledge sources.

B. Alawaji, M. Hakami, and B. Alshemaimri [6] investigates the use of generative language models for automating user story categorization in software development. The authors evaluate GPT-3.5-turbo and Llama-2-chat-hf models

using zero-shot and few-shot prompting, showing strong performance, especially with zero-shot prompting. The paper suggests future work exploring more models and prompt techniques and incorporating user feedback into the evaluation process.

Alamelu et al. [7] presents a web application that uses Natural Language Pro-cessing (NLP) to automate resume screening and ranking. The system extracts and compares relevant information from resumes to job descriptions, scoring candidates based on the best match. This automation streamlines recruitment, reducing manual review time and minimizing human biases in the hiring process.

Albassam [11] showcases how AI is rapidly transforming recruitment, offering powerful tools like resume screening, candidate matching, video interviewing, chatbots, predictive analytics, gamification, VR assessments, and social media screening. These technologies promise increased efficiency, reduced bias, and better hiring outcomes. However, they also present challenges such as algorithmic bias, potential negative impact on candidate experience, and privacy concerns, necessitating human oversight. Further research is crucial to explore emerging technologies, mitigate bias, address legal and ethical implications, and understand the evolving impact of AI on recruitment. HR professionals, in turn, need to acquire new skills to effectively harness these powerful tools.

A review of research on resume processing and job matching reveals several gaps. First, there is a lack of comparative analysis of different methodologies. The potential of fine-tuning large language models (LLMs) to improve accuracy and performance in resume processing has not been fully explored. While retrieval-augmented generation (RAG) techniques have been studied, their application in recruitment—especially factors like retrieval speed and scalability, which are essential for real-time job matching—remains under-examined. Current approaches primarily focus on Named Entity Recognition (NER) and section seg-mentation methods. While gender bias in AI-driven recruitment has been ad-dressed, other demographic biases require further investigation to ensure fair-ness. When the accuracy of the extraction and matching process is being focused on, the efficiency is proportionally reduced. Additionally, the reliance on outdated training datasets, often based on short resumes, limits models' ability to handle the complexity of modern resumes. Addressing these gaps will significantly im-prove AI-driven recruitment, leading to more accurate, efficient, and equitable resume processing and job matching.

## III. Background

With the digitalization of recruitment, traditional methods like manually re-viewing resumes and matching candidates to job descriptions have become inefficient and time-consuming. The growing volume of job postings and candidates requires more automated solutions. Artificial Intelligence (AI) and Natural Language Processing (NLP) can automate the extraction of key information from resumes, improving candidate-job matching. Recent advancements in Large Language Models (LLMs) enable efficient parsing of complex resume data, extracting relevant details like skills, education, and experience. When combined with Retrieval-Augmented Generation (RAG) techniques, these models improve the accuracy of matching candidates to job requirements, enhancing both speed and efficiency in the recruitment process.

### *A. Large Language Models*

Large Language Models (LLMs) are a type of artificial intelligence (AI) that uses deep learning techniques to understand and generate human language. These models are typically trained on vast amounts of text data from books, websites, articles, and other sources to learn patterns in how words, sentences, and even entire conversations flow. The more data they are trained on, the better they can predict and generate coherent, contextually appropriate responses.

Mistral AI's Mixtral-8x7B-Instruct-v0.1 model is a specialized large language model (LLM) with 8.7 billion parameters, which are the adjustable weights that enable the model to process and generate language. What distinguishes Mixtral-8x7B-Instruct is its fine-tuning for instruction-following tasks. It is optimized not just for text generation, but to follow complex user instructions with high accuracy. Whether tasked with summarizing a document, answering questions, or providing step-by-step guidance, Mixtral-8x7B-Instruct excels in these tasks.

However, despite being a Mixture of Experts (MoE) model, its context-free approach limits its full potential, preventing it from consistently delivering accurate results, especially as a relatively new model. This paper utilizes Mixtral-8x7B-Instruct, acknowledging its novelty and exploring it to a greater extent. Microsoft’s Phi-3 series models could also be considered, as they have demonstrated strong performance in similar tasks.

### *B. Retrieval-Augmented Generation*

Retrieval-Augmented Generation (RAG) is an advanced technique used to enhance the capabilities of large language models (LLMs) by integrating them with external information sources, such as databases, search engines, or document collections. Unlike traditional LLMs, which generate responses based solely on the knowledge encoded during training, RAG models [8] dynamically retrieve relevant pieces of information from external sources in real-time to enrich their responses. This retrieval process allows the model to draw on more specific, up-to-date, or specialized knowledge that might not be captured during training and it helps give context to an LLM.

### *C. Prompt Engineering*

Prompt engineering refers to the practice of crafting and refining input prompts to guide large language models (LLMs) in producing desired outputs. Since LLMs rely on the prompts they receive to generate responses, how a prompt is framed can have a significant impact on the quality, relevance, and accuracy of the generated content. The goal of prompt engineering is to make the interaction with the model more efficient and effective, especially when the model is being used for complex or specialized tasks. Techniques in prompt engineering [9][10] often involve clearly specifying instructions, providing relevant context, or refining the output style. For example, asking a model to “summarize this text in one paragraph” or “generate a step-by-step solution” makes the task clear and targeted. Providing context such as background information or key details, can also help the model generate more accurate responses by anchoring its understanding to the task at hand.

## IV. Proposed Work

The primary focus of this work is to improve the accuracy of extracting relevant details from both resumes and job descriptions, while maintaining system efficiency. The basic workflow of the job portal is illustrated in Fig. 1. An employer or recruiter inputs a job description and the system extracts key information, such as required skills, work experience and educational qualifications. Similarly, information is extracted from candidates' resumes. These extracted data are then used for matching candidates to the job openings.

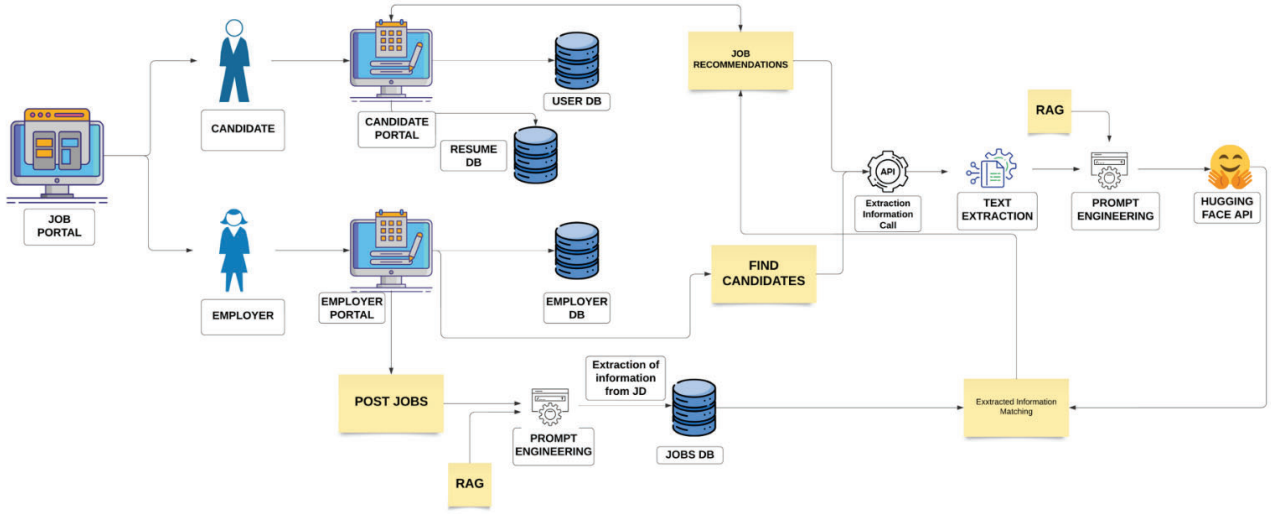


Fig. 1. Overall System Architecture for the Job Portal

The extraction of vital information is the most crucial step, as job descriptions and resumes can be provided in a variety of formats. Instead of relying solely on Named Entity Recognition (NER)-based models or rule-based approaches, this work incorporates prompt engineering and Retrieval-Augmented Generation (RAG) techniques to enhance the extraction process. The goal of this paper can be divided into three main modules:

1) Retriever module to generate the context.

2) Extraction of relevant information by augmentation and generation.

3) Matching algorithm for the extracted information.

### A. Retriever Module

The first component being addressed in this work is the retriever module, which generates the context for the matching process. The job description or resume text is first used as a query in a TF-IDF (Term Frequency-Inverse Document Frequency) model, along with two datasets: a skills dataset for context on relevant skills and a degree dataset for context on educational qualifications, as shown in Fig. 2. These texts are compared to the datasets to retrieve the top three skill stacks and the top two educational degree stacks, which are then sent to the LLM as context for matching.

"Full-Stack Python Flask Django JavaScript React.js Node.js HTML CSS MySQL PostgreSQL MongoDB REST APIs GraphQL Docker Kubernetes AWS Git"
"Backend Python Java Springboot Flask Django Node.js MySQL MongoDB PostgreSQL NoSQL Redis Kafka AWS EC2 Lambda Microservices Docker APIs Elasticsearch SQL RDBMS"
Frontend HTML CSS JavaScript React.js Angular Vue.js Bootstrap Tailwind CSS jQuery AJAX Webpack TypeScript
"Database SQL MySQL PostgreSQL MongoDB Cassandra Redis Oracle NoSQL RDBMS Data Warehousing ETL Snowflake BigQuery"
"Cloud AWS Microsoft Azure Google Cloud Platform Kubernetes Docker EC2 S3 Lambda Redshift CloudFormation Serverless Jenkins Terraform"
"DevOps Docker Kubernetes Jenkins Terraform CI/CD AWS Google Cloud Ansible Helm GitLab CI GitHub Actions Infrastructure as Code ELK Stack"
"Data Scientist Python R TensorFlow Keras Scikit-learn Pandas NumPy Matplotlib Seaborn Apache Spark Jupyter Notebooks PyTorch H2O.ai Big Data LightGBM"
"UI/UX Figma Sketch Adobe XD InVision Photoshop Illustrator Prototyping Wireframing Interaction Design Usability Testing User Research Information Architecture"
"Project Management Agile Scrum Waterfall Kanban Jira Trello Asana Lean Six Sigma Risk Management Resource Allocation Time Management Leadership"
"Cybersecurity Network Security Cryptography Firewalls VPN IDS IPS Penetration Testing Vulnerability Scanning Data Encryption Cloud Security Security Audits"
"Big Data Hadoop Apache Spark Kafka HBase NoSQL Cassandra MapReduce BigQuery Hive AWS EMR Data Lakes Distributed Computing"
"Business Intelligence Tableau Power BI SQL ETL Data Warehousing BigQuery Redshift Snowflake Google Data Studio Alteryx Trifacta Apache Airflow"

Fig. 2. Sample technical skills dataset for retriver module

Currently, the context is generated using a small dataset created in Python, which still provides good accuracy for technical job descriptions and resumes. The effectiveness of the context improves as the dataset is expanded. One of the key advantages of using this approach, combined with the TF-IDF method, is its ability to handle the introduction of new skills into the job market. When a new skill emerges, only the relevant technological stacks need to be updated, rather than requiring a full model retraining. This dataset was specifically created for this work, with a focus on technological skills, rather than being scraped from external websites. The overall flow of the retriever module is illustrated in Figure 3, where the arrows point to the dataset (shown in the last cell). This is because the cosine similarity function returns the top indices, which require a lookup to identify the corresponding skill and degree stacks.

Pseudo-Code for retrieving context using python is given as follows:

```
documents=dataset + [JD_or_Resume]

vectorizer=TfidfVectorizer(stop_words='english', max_df=0.95, min_df=2)

tfidf_matrix=vectorizer.fit_transform(documents)

cosine_similarities=cosine_similarity(tfidf_matrix[-1], tfidf_matrix[:-1])

top_indices=cosine_similarities.argsort()[0, -3:][::-1]

top_skills=[skill_sets[i] for i in top_indices]
```

The max_df and min_df parameters are crucial as they control the importance of words in the document. For instance, a skill like "Java" might be repeated several times in a job description, but its significance is high. These parameters help

adjust for such variations, ensuring the most relevant terms are prioritized.

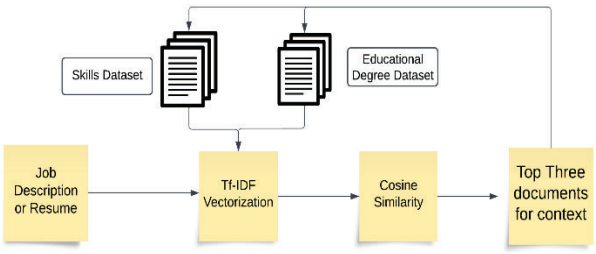


Fig. 3. Retriever Module Flowchart

### B. Augmentation and Generation

The next step in the process is augmentation, which requires the engineering of a well-structured prompt. The design of the prompt varies depending on the model being used, as each model is trained differently and can interpret inputs to varying degrees. For the Mistral AI model tested in this work, a carefully crafted prompt has been developed to generate accurate responses.

Prompt - Extract using {context} but do not copy paste the relevant skills, Education degree and work experience in months/years from given job description. If no skill or no work experience or no education degree then mention N/A: {job_desc} Output format: Skills: [As a list] Education Degree: Work Experience in number and years or months:

This prompt, which incorporates the generated context, is referred to as an augmented prompt. It is sent to the LLM through the Hugging Face API, or the model can be downloaded and run locally if the computational resources are sufficient for the generation task. With the provided context, the LLM can identify the relevant skills, educational degrees and work experience from the job de-scription or resume.

### C. Matching of Extracted Information

The information extracted from the job description and resumes, using the methods outlined in the previous sections, is matched using a simple weighted sum-based approach. The matching process involves comparing the skills, education and work experience between the job description and the candidate's resume. For skills, the overlap between the extracted skills from both documents is assessed. In the case of education, the comparison is made between the degrees listed in the job description and the resume and if necessary, Levenshtein distance is applied to determine the similarity between degree titles. For work experience, the candidate's experience is compared to the job description; if the candidate's experience exceeds the job requirement, the difference is factored in by adding more weight to this aspect. Since the same LLM and retriever model are used for both the job description and the resume, the structure of the extracted information remains consistent, which ensures that the matching process is reliable and accurate.

## V. Results and Discussion

The effectiveness of the proposed model is validated using two metrics – speed of the model and the relevance of the extracted data (measured in terms of precision).

### A. Speed

Table 1 shows the average processing time for various steps in the system. The data is based on multiple samples of job descriptions and resumes, with the over-all processing time averaging less than 15 seconds, even when making calls to the Hugging Face Model API for the generation part, assuming an average internet connection.

TABLE I. Speed of Mistral AI model

| Serial No. | Component | Speed |
|---|---|---|
| 1 | Context Retrieval using TF-IDF | 0.2s |
| 2 | Prompt Generation using Mistral AI model | 4-9.5s |
| 3 | Full Process including formatting | <15s |

### B. Relevance of the Extracted Data

Results of a context-free prompt versus a prompt with context are compared using two example job descriptions. For a sample job description taken from LinkedIn from Sopra Steria for a Junior data engineer role, a context free output would be as shown in Figure 4. As it can be seen excess or unnecessary data has been extracted in a few places such as “ETL tools like design and maintain data pipelines using Azure Factory for seamless data integration and migration”, “SQL tools for ETL processes” and this output is not consistent.

To evaluate the quality of skill extraction, precision is a useful metric. Precision measures how accurately skills have been extracted and it is defined by (1).

$$Precision = \frac{True\ Positives}{Positives + False\ Positives} \quad (1)$$

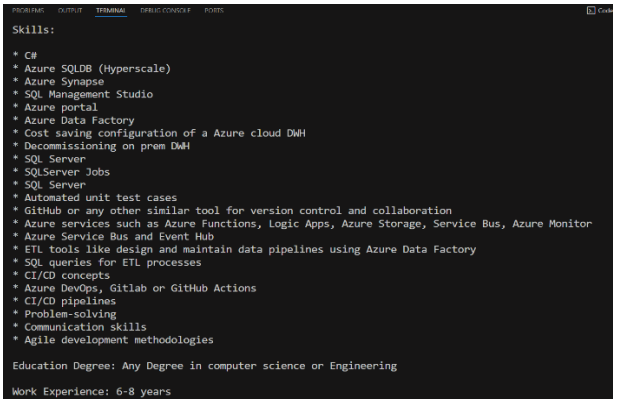


Fig. 4. Extracted information from a sample job description with no context

To resolve this issue, RAG-based context retrieval is employed. For the same job description, relevant skills are extracted more accurately and precisely, as demonstrated in Figure 5. This process begins by calling the retrieval module to gather context, which is then used to augment the prompt.

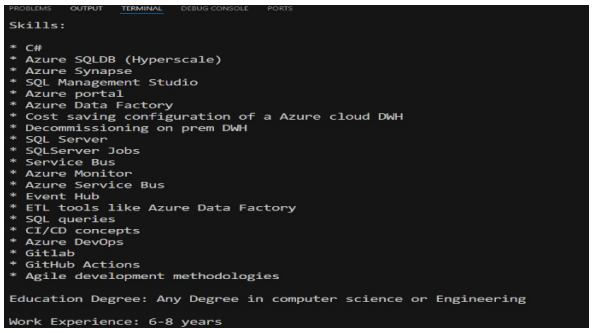


Fig. 5. Extracted information from the same sample job description with context

Table 2 presents the extracted information categorized as either relevant or irrelevant. The condition for irrelevant information includes redundancy or skills that contain too much noise (such as stop words). Using Equation (1), the precision for information extracted without context is calculated to be 73.9%. When extraction is performed with context, the precision improves to 90.48%. This demonstrates a significant improvement in the accuracy of the extracted information when context is taken into account.

As it can be seen using RAG with prompt engineering significantly improved the precision of the data extracted from an unstructured job description. RAG techniques—specifically the retrieval of relevant context—are crucial in providing the LLM with meaningful input. Here, context refers to the technological and degree stacks most closely aligned with the job description or resume. This approach reduces the workload for both recruiters and candidates by automating much of the process. The system's accuracy is significantly improved when context is provided, compared to the context-free approach.

TABLE II. CATEGORIZATION OF EXTRACTED INFORMATION

| Serial No. | Component | Count |
|---|---|---|
| 1 | Relevant Information without Context | 17 |
| 2 | Irrelevant Information without Context | 23 |
| 3 | Relevant Information with Context | 19 |
| 4 | Irrelevant Information with Context | 21 |

A key advantage of using RAG-based prompt engineering is that it allows multiple LLMs especially open source LLMs to be used without the need for fine-tuning. Instead, only the prompt requires engineering, reducing overall development time and complexity. Additionally, the RAG component ensures the system remains up to date with new skills, enhancing its relevance in the fast-evolving job market. While the examples demonstrated in this paper focused on technical skills and job descriptions, this approach can be easily adapted to other domains or disciplines by modifying the dataset and augmenting the retriever with relevant stacks.

## VI. CONCLUSION

This paper highlights the potential of integrating advanced techniques like Retrieval-Augmented Generation (RAG) and prompt engineering to enhance the efficiency and accuracy of resume and job description matching. By addressing key gaps identified in current research, such as the underexplored impact of fine-tuning large language models (LLMs) and the need for real-time retrieval speed, this approach provides a more robust solution to automate and improve recruitment processes. The proposed system effectively uses RAG techniques for context retrieval, ensuring that the extracted information is both relevant and accurate, thereby improving job matching accuracy with no bias of region or gender. The integration of prompt engineering optimizes data extraction, streamlining the process for both recruiters and candidates and gives results in less than 15 seconds without compromising accuracy and precision. While challenges like handling very unstructured resumes with pictures remain, the proposed system offers a significant improvement in reducing manual effort and ensuring more precise matching.

## REFERENCES

[1] S. Tyagi and A. Anuj, "Promoting Gender Fair Resume Screening Using Gender-Weighted Sampling," in Proc. Int. Conf. Computing, Machine Learning and Data Science (CMLDS '24), New York, NY, USA, 2024, pp. 1–5.

[2] K. V. Deshpande, S. Pan, and J. R. Foulds, "Mitigating Demographic Bias in AI-based Resume Filtering," in Adjunct Proc. 28th ACM Conf. User Modeling, Adaptation and Personalization (UMAP '20 Adjunct), Genoa, Italy, 2020, pp. 1–8.

[3] T. Schmitt, P. Caillou, and M. Sebag, "Matching Jobs and Resumes: A Deep Collaborative Filtering Task," in Proc. 2nd Global Conf. Artificial Intelligence (GCAI 2016), Berlin, Germany, September 2016, pp. 1–5. [Online]. Available: hal-01378589.

[4] Z. Hu, C. Wang, Y. Shu, H.-Y. Paik, and L. Zhu, "Prompt Perturbation in Retrieval-Augmented Generation Based Large Language Models," in Proc. 30th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD '24), Barcelona, Spain, 2024, pp. 1–12.

[5] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, "A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models," in Proc. 30th ACM SIGKDD Conf.

[6] B. Alawaji, M. Hakami, and B. Alshemaimri, "Evaluating Generative Language Models with Prompt Engineering for Categorizing User Stories to its Sector Domains," in 9th IEEE Int. Conf. for Convergence in Technology (I2CT 2024), Pune, India, 2024, pp. 1–8.

[7] M. T. Alamelu, D. Kumar, R. K. Sanjana, J. Sree, A. S. Devi, and D. Kavitha, "Resume Validation and Filtration using Natural Language Processing," in 2021 10th Int. Conf. Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON), 2021, pp. 1–5.

[8] A. Salemi and H. Zamani, "Evaluating Retrieval Quality in Retrieval-Augmented Generation," in Proc. 47th Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '24), Washington, DC, USA, 2024, pp. 1–6.

[9] D. Park, G. An, C. Kamyod, and C. G. Kim, "A Study on Performance Improvement of Prompt Engineering for Generative AI with a Large Language Model," J. Web Engineering, vol. 22, no. 8, pp. 1187–1206, 2024.

[10] F. Polat, I. Tiddi, and P. Groth, "Testing Prompt Engineering Methods for Knowledge Extraction from Text," Semantic Web, vol. 1, no. 0, pp. 1–4, 2023.

[11] W. A. Albassam, "The Power of Artificial Intelligence in Recruitment: An Analytical Review of Current AI-Based Recruitment Strategies", *J. Professional Business Review*, vol. 8, no. 6, p. e02089, Jun. 2023.