Proceedings of the 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI-2024)
IEEE Xplore Part Number: CFP24VL6-ART; ISBN: 979-8-3503-8960-9

2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI) | 979-8-3503-8960-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDICI62993.2024.10810811

A RAG based Personal Placement Assistant System using Large Language Models for Customized Interview Preparation

Samay Patel
Computer Science & Engineering Department
Devang Patel Institute of Advance Technology and Research (DEPSTAR), Charotar University of Science and Technology (CHARUSAT)
Anand, India
samaypatel0402@gmail.com

Jeet Patel
Computer Science & Engineering Department
Devang Patel Institute of Advance Technology and Research (DEPSTAR), Charotar University of Science and Technology (CHARUSAT)
Anand, India
jeet812patel@gmail.com

Dhairya Shah
Computer Science & Engineering Department
Devang Patel Institute of Advance Technology and Research (DEPSTAR), Charotar University of Science and Technology (CHARUSAT)
Anand, India
dtshah24022006@gmail.com

Parth Goel
Computer Science & Engineering Department
Devang Patel Institute of Advance Technology and Research (DEPSTAR), Charotar University of Science and Technology (CHARUSAT)
Anand, India
parthgoel.ce@charusat.ac.in

Bankim Patel
Computer Science & Engineering Department
Devang Patel Institute of Advance Technology and Research (DEPSTAR), Charotar University of Science and Technology (CHARUSAT)
Anand, India
bankimpatel.dcs@charusat.ac.in

Abstract*— This paper introduces a Personal Placement Assistant (PPA) framework that utilizes advanced Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs) to automate and personalize job placement preparation. The system integrates Natural Language Processing (NLP) techniques, including text embedding using the all-MiniLM-L6-v2 transformer model and semantic retrieval using ChromaDB for accurate resume analysis and context-aware question generation. The PPA is structured into three core components: the Retriever, using PyMuPDF for resume parsing and recursive text chunking for efficient vector storage and search; the Analyzer, employing the Google Gemini-1.5-flash model for domain extraction and percentage-based content profiling; and the Generator, which produces domain-specific MCQs, coding challenges, and interview questions aligned with Bloom's Taxonomy. RAG enhances the system's ability to integrate external knowledge, improving the contextual relevance of the generated content. Evaluation results demonstrate an 83.77% accuracy in domain-specific extraction and question generation, confirming the PPA's effectiveness in automating personalized job preparation across industries.

Keywords*—Retrieval Augmented Generation (RAG), Large Language Models (LLMs), Transformers, Personal Placement Assistant (PPA), ChromaDB, Interview Preparation

I. INTRODUCTION

The job market has undergone significant transformation in recent years, with the global workforce facing unprecedented challenges. According to the World Economic Forum's Future of Jobs Report 2023 [1], 23% of jobs are expected to change in the next five years. Additionally, LinkedIn's Global Talent Trends [2] report indicates that 89% of talent professionals believe soft skills are increasingly important in the hiring process. These shifts, coupled with rapid technological advancements, have created a complex landscape for job seekers. Traditional placement preparation systems often rely on static databases and rigid algorithms, resulting in generic question sets and limited adaptability to individual candidate profiles or rapidly evolving industry requirements. These systems typically offer pre-defined question banks that fail to account for the nuanced skills and experiences of each applicant. Thus, introducing a AI-driven Personal Placement Assistant that uses RAG [3] and LLMs [4] enabling real-time semantic analysis of resumes, extracting and contextualizing key resources. This detailed understanding enables the dynamic generation of customized assessment materials, perfectly adjusted to each candidate's expertise level and aligned with current industry-specific skill demands. This approach addresses the challenges faced by candidates in obtaining personalized guidance and accessing relevant, well-structured questions aligned with placement processes. This solution brings everything together in one place, reducing the need for students to use different platforms and making sure the preparation materials

are customized specifically to each student's skills and the requirements of the job market

As chatbots increasingly employ NLP techniques, such as NLTK for Python, to analyse speech and deliver responses that copy human interaction, there is potential to design systems that more effectively guide candidates through job preparation [5]. Moreover, recent advancements in AI and NLP [6] have highlighted opportunities to enhance conventional job preparation systems, particularly in the areas of context-aware retrieval and RAG integration. While existing approaches have advanced keyword extraction and question generation, often falling short in providing personalized guidance, leaving candidates uncertain about optimal preparation strategies. This limitation is worsen by fragmented resources across multiple platforms and questions that frequently misalign with actual interview processes and Bloom's Taxonomy cognitive levels, particularly in application, analysis, and evaluation. Biancini et al. advanced the field by exploring large language models for question generation in educational contexts, opening paths for application in comprehensive resume parsing [7]. The continuous development of NLP technology is significantly enhancing job preparation platforms, allowing for more personalized and effective support. Building upon these foundations, our project integrates the Google Gemini LLM with context-aware retrieval RAG system, facilitating refined resume analysis and personalized question generation. This approach ensures precise extraction, semantic understanding, and the generation of contextually enriched & domain-specific questions, providing a comprehensive, adaptive preparation tool customized to individual candidates' profiles and evolving job market demands.

This study presents a novel framework that integrates advanced NLP techniques with a state-of-the-art LLM, specifically the Google Gemini model [8], in conjunction with RAG to address critical challenges in context-aware information retrieval and personalized content generation. By implementing RAG, the model's ability to dynamically incorporate external knowledge, significantly improving the contextual relevance of generated outputs. This approach enables the generation of well-grounded diverse questions aligned with Bloom's Taxonomy, ranging from analytical MCQs to complex coding challenges. Through the collective integration of these advanced NLP methodologies, our system aims to provide a more robust, adaptive solution for automated job preparation, effectively mitigating the shortcomings of traditional keyword-based and non-contextual approaches present in current systems.

The main contributions of this work can be summarized as follows:

- To address the challenges of limited contextual understanding and personalization in existing systems, enabling robust and adaptive resume analysis and personalized recommendations for interview preparation.
- Utilizing vector similarity retrieval and the Gemini-1.5-flash LLM model, the presented RAG approach integrates external knowledge to significantly increase the efficiency and contextual relevance of automated resume-oriented interview question generation which assist in placement.

II. Related Work

The following section explores various studies relevant to the domain of resume parsing, summarization, and automated question generation, highlighting their methodologies, achievements, and how this research builds upon these foundations.

Malinen and Esko proposed an interactive document summarizer using LLM technology [9]. This master's thesis explores into the properties and usage of LLMs and generative AI, specifically focusing on RAG. The thesis includes the development of a software application capable of interactive discussions within the context of given documents using modern development tools and environments. The main findings demonstrate the application's ability to summarize and interact with documents effectively. However, the work primarily focuses on general document summarization and interaction, rather than targeting resume-specific content or the generation of domain-specific questions. Similarly, Saba et al. proposed a method for summarizing electronic health records using retrieval augmented generation and question-answering with large language models [10]. Although focused on health records, this research is relevant due to its use of LLMs and retrieval augmented generation, similar to the techniques used in the present project for resume analysis. The main findings indicate significant improvements in the accuracy and relevance of summaries, demonstrating the effectiveness of integrating question-answering techniques with LLMs in generating precise and useful summaries. Chataut et al. conducted a comparative study of domain-driven terms extraction using large language models [11]. This research focused on the extraction of domain-driven terms using LLMs, comparing different models' effectiveness in identifying relevant terms within specific domains. The results demonstrated that domain-specific vocabulary plays a key role in improving both the accuracy and relevance of information retrieval and summary generation tasks. While significant insights into term extraction are provided, the study does not integrate this capability into a broader framework for comprehensive resume analysis and automated question generation. Furthermore, Varalakshmi and Bugatha proposed an AI-powered system that extracts keywords from resumes and job descriptions to generate relevant Q&A [12]. The system achieved a 97% accuracy rate in generating meaningful Q&A, significantly improving the possibility of finding suitable job matches and outperforming prior models. Despite its high accuracy, the approach focuses primarily on Q&A generation for interview preparation rather than providing a detailed summarization and domain analysis of resume content, which are crucial for a comprehensive understanding of a candidate's skills and qualifications. Moreover, Elmessiry et al. also explored the evolution of AI towards Education-Specific Retrieval Augmented Generative AI (ES-RAG-AI) in their paper [13]. The research addresses the challenges of verification and transparency in AI-

generated responses within educational settings. The main finding relevant to this research is the introduction of ES-RAG AI, which aims to tailor AI capabilities to meet educational needs while ensuring accountability and authenticity. This approach aligns with the goal of enhancing the transparency and reliability of AI-generated content in educational tools. Sajid et al. proposed a study which focuses on improving the e-recruitment process through an advanced resume parsing framework [14]. The study compares the performance of Llama 2 [15], Mistral [16], and GPT- 3.5 [17]in creating MCQs, highlighting GPT-3.5 as the most effective model. The main finding relevant to this research is the potential of LLMs to generate high-quality educational content, particularly MCQs, which can enhance the learning experience. Traditional methods for resume information extraction face challenges due to the variety of resume formats and the need for large annotated datasets. The main finding relevant to this research is the framework's ability to handle diverse resume formats and enhance the accuracy of information extraction, which is crucial for developing a reliable and efficient resume analysis tool. Hasan et al. worked on implementing Automatic Question Answer Generation (AQAG) using a fine-tuned generative LLM to enable the creation of diverse academic questions [18]. Their findings indicate that employing unsupervised learning methods and fine- tuning with the RACE dataset [19] significantly enhances the efficiency of generating multiple types of questions, such as MCQs, conceptual, and factual questions. This research is useful for our study as it demonstrates the potential of generative LLMs in automating the question-generation process, thereby improving the scalability and consistency of educational assessments. While these studies have significantly advanced the field and laid a foundation for further research, a literature review reveals several persistent gaps in the current study. Many studies focus on isolated aspects of text processing, such as document summarization, classification, or domain-specific term extraction, without integrating these elements into comprehensive frameworks. There is a notable lack of research on personalized content generation that adapts to individual document characteristics. Additionally, while LLMs show promise in various applications, their full potential in specialized document analysis and customized content generation remains less explored. Furthermore, there is lack in generation of contextually enriched content using LLMs.

III. Methodology

The methodology integrates resume analysis with targeted question generation, ensuring that the recruitment process is both efficient and personalized. By aligning questions with the candidate's specific skills and experiences, the system provides more relevant and tailored assessments that enhance the accuracy and effectiveness of the interview process. The Figure 1 illustrates our approach, starting with extracting textual content from resumes using PyMuPdf [20] that maintain document integrity. Next, the embedding generation phase converts this text into high-dimensional vector representations via hugging-face all-MiniLM-L6-v2 [21] transformers model, capturing semantic differences for precise, context-aware

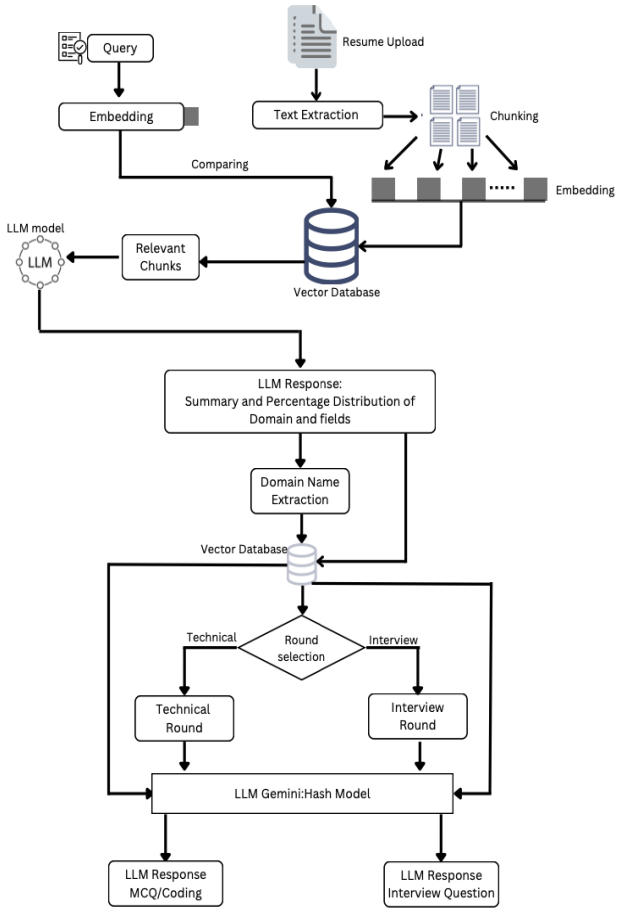

Fig. 1. Architecture of AI-powered resume analysis and customized interview generation system

analysis. These embeddings are stored in a vector database using ChromaDB [22], enabling efficient retrieval and comparison. The Analyser component processes this data to generate a detailed summary and domain-wise percentage distribution, providing insight into the candidate's expertise. Finally, the Generator component leverages this analysis to create domain-specific questions: MCQs and coding challenges for the technical round, and questions assessing problem-solving, leadership, and cultural fit for the interview round.

3.1 The Retriever

The Retriever component is responsible for processing the candidate's resume and preparing it for analysis. This component consists of several crucial steps: document pre-processing, text extraction, embedding generation, and vector storing.

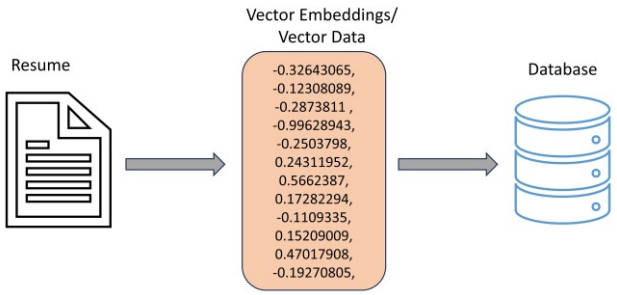

Fig. 2. Vector embedding process for resume data

3.1.1 Document Pre-processing

In a RAG system, the first crucial step is getting the documents ready for efficient retrieval and analysis. This groundwork involves organizing the knowledge base to make sure information can be pulled out accurately and swiftly. To start, the system gathers all the text from the Resume file uploaded by users. This collected text is then converted into a simple, uniform format to make sure all types of documents are handled the same way. Next comes the chunking process, which uses the Recursive-Character-Text-Splitter to break the text at natural points like the end of paragraphs or sentences, employing a chunk size of 1000. This approach makes the whole process of finding and using information much more efficient.

3.1.2 Text Extraction

To process Resume, a python library PyMuPDF is utilized in the text extraction phase. This library enables accurate and efficient extraction of text content from uploaded resumes. The extraction process precisely scans each page of the PDF document, capturing all textual elements while maintaining their structural integrity. This approach ensures the complete retrieval of all data, including text embedded in different layouts, columns, and formatting styles frequently seen in resumes. The extracted text is then further processed to remove any remaining formatting artifacts, resulting in a clean, plain text representation of the resume's content. This refined text serves as a standardized input for the system's embedding generation and analysis phases, forming a reliable foundation for further analysis.

3.1.3 Embedding Generation

After text extraction, the system generates embeddings to transform the textual data into high-dimensional vector representations that captures semantic meanings shown in Figure 2. This system employs Hugging Face sentence-transformers model all-MiniLM-L6-v2 embedding model, known for its balance of performance and computational efficiency, which further enhances the process by creating high-quality vector representations for precise and efficient analysis. The embeddings capture the contextual tones of the resume content, facilitating robust semantic similarity searches, enhancing the precision and efficiency of subsequent matching and retrieval operations.

3.1.4 Vector Storing

Once Embeddings are generated, it is effectively stored in a vector database. In order to facilitate fast retrieval and comparison of resume data, this step is essential for maintaining an organized repository. This system employs Chroma-DB as a vector storage system, enabling swift similarity comparisons and efficient data retrieval. It can store vectors with additional metadata and allows for filtering during the query search on the vector database. This scalable architecture ensures the rapid storage and retrieval of numerous resume embeddings.

3.1.5 Prompt Engineering

Prompt engineering is the process of crafting and refining prompts to obtain specific outputs from AI models. It is essential for formulating queries that help generative AI models understand the intent and complexity behind the input, beyond just the language. This system prompts generates questions for both technical and interview rounds. Technical prompts create MCQs targeting analysis, application, and evaluation, and coding challenges reflecting real-world scenarios. Interview prompts generate questions assessing problem-solving, leadership, communication skills, and cultural fit, customized to resume analysis.

The key elements of a prompt include:

- Instruction**:** This is the primary directive that guides the model on the specific task it needs to perform.
- Context**:** This adds relevant details to help the model understand the broader scenario or background, aiding in generating better results.
- Input Data**:** This is the specific information the model needs to process.
- Output Indicator**:** This component directs the model on the expected type or format of the response.

3.2 The Analyzer

The Analyzer component executes a comprehensive evaluation of the candidate's resume by employing a RAG based system with LLM. It thoroughly processes and synthesizes extracted resume data to produce an in-depth analysis, emphasizing key credentials, experiences, and areas of expertise. This approach ensures a thorough and accurate assessment of the candidate's profile.

3.2.1 Summary Generation

The Analyzer employs the Google's Gemini-1.5-flash model to create a structured summary of the resume using RAG system. This involves parsing the resume text to identify and aggregate key skills, interests, and areas of expertise into a clear overview. By synthesizing relevant qualifications and experiences, the model provides a high-level combination of the candidate's professional profile. This RAG-based approach ensures a clear and concise summary, enhancing the effectiveness and efficiency of subsequent analyses.

Prompt: {"An expert resume analyst. Analyze the following resume text and provide the following:\n\n"
"1. A detailed percentage distribution of fields/domains present in the resume, with keywords extracted from each domain.\n"
"2. Identify and extract standard domain names relevant to the candidate's expertise using semantic analysis techniques.\n"
"3. Generate a comprehensive distribution profile that highlights the relative prominence of each domain, including the associated keywords.\n"
"\nHere is an example of how the output should look:\n\n"
"### Percentage Distribution of Fields/Domains\n"
"Note: only include standard technologies as keywords.\n"
"- Machine Learning (ML): 40%\n"
" - Keywords: Algorithms, Keras, PyTorch, Scikit-Learn, Predictive Models\n"
"- Data Science (DS): 30%\n"
" - Keywords: Data Analysis, Pandas, NumPy, Visualization, Statistical Methods\n"
"- Software Development (SD): 30%\n"
" - Keywords: Python, Java, Software Engineering, APIs, Git\n"}

3.2.2 Domain Name Extraction- Percentage Distribution with Keywords

The Analyzer employs RAG-based system followed by LLM prompt which specifically assess resume content, focusing on the distribution of specialized fields and domains. By extracting domain-specific keywords, the system determines their relative importance and frequency within the resume, resulting in a detailed distribution profile that accurately reflects the candidate's domain expertise.

The Analyzer also utilizes the Google's Gemini-1.5-flash model for semantic and pattern-based extraction of domain names, employing contextually-aware generation and categorize terms that align with the candidate's professional skill set. This integrated RAG-based approach enhances the precision and contextual relevance of the generated questions, ultimately refining the robustness and depth of the analytical outputs.

This approach uses a well-designed formula which represent how LLM would calculate the percentage distribution. The process begins by calculating the frequency of domain-specific keywords within the resume text, denoted as $F(D_i, T)$ Each domain D_i is associated with a set of keywords K_{ij} which suggests the candidate's expertise in that domain.

Let:

- T denote the total text in the resume.
- D_i represent Domain i (e.g., Machine Learning, Data Science)
- K_{ij} be the keyword j associated with Domain D_i.
- $F(K_{ij}, T)$ denote the frequency of occurrence of keyword K_{ij} *within* the text T.
- $P(D_i)$ be the percentage distribution of Domain D_i.

$$F(D_i, T) = \sum_{j=1}^{n_i} F(K_{ij,} T)$$

By aggregating these keyword frequencies, the formula provides a quantitative measure of how significantly each domain is represented in the resume. This frequency is then normalized against the total frequency of all identified domains in the resume, resulting in a percentage distribution $P(D_i)$

$$\mathrm{P(D_i)} = \frac{\mathrm{F(D_i,T)}}{\sum_{\mathrm{k=1}}^{\mathrm{m}} \mathrm{F(D_k,T)}}$$

Here, m is the total number of domains identified within the resume.

Once the percentage distribution is calculated, the RAG-based LLM uses this information when generating Questions. For example, if the domain "Machine Learning" has the highest percentage distribution, the LLM will emphasize this domain in its generated questions. The formula's integration with RAG framework represents a significant advancement in the field of automated resume analysis. By providing a strong mechanism for domain extraction and content profiling, it enhances the accuracy, relevance, and accountability of the LLM's outputs.

Prompt:

{"An expert resume analyst. Analyze the following resume text and provide the following:\n\n"
"1. A brief summary of the resume, including the candidate's main interests and the fields the candidate is most passionate about\n."
"2. Identify key competencies, interests, and areas of expertise, and synthesize them into a structured summary.\n"
"Here is an example of how the output should look like:\n\n"
"### Summary\n
"The resume indicates a strong background and interest in machine learning, data science, and software development. "
"The candidate has worked on several projects involving machine learning algorithms, data preprocessing, and building software applications. "
" The candidates have demonstrated proficiency in Python, Java, and various machine learning frameworks. The candidate is passionate about solving complex problems using AI and has a keen interest in continuing to develop their skills in this area.\n\n"}

3.3 The Generator

The Generator component is crucial in producing customized, domain-specific questions based on the analysis of the resume. It generates MCQs, coding challenges, and interview questions, ensuring each question is relevant and challenging. The Generator creates precise, context-aware questions customized to the skills and experience, using advanced prompt engineering and RAG techniques, This innovative approach differentiates our system by delivering highly targeted and contextually relevant questions, significantly enhancing the preparation and assessment process. Additionally, this system implements a dynamic few-shot learning approach, incorporating example questions to guide the model's output.

3.3.1 MCQs Generation

Developing MCQs across different difficulty levels within a domain requires careful planning to effectively assess varying levels of knowledge and skills. The system uses a well-designed prompt to specify the format, structure, and context matter of the questions. The prompt directs the generation of MCQs covering various intellectual skill levels according to Bloom's Taxonomy, ensuring an assessment from basic knowledge recall to higher-order thinking skills such as application and analysis. The generated MCQs are designed to match the experience and areas of expertise, ensuring relevance and alignment with the professional profile. This approach guarantees that the questions are not only diverse but also relevant to the candidate's specific skills and knowledge.

Following is the prompt used for MCQs generation:
“““*Subject:* {selected_domain}
MCQ Prompt::Generate {num_questions} code snippet and practical knowledge types of multiple-choice questions (MCQs) for the subject of {selected_domain} aligned with the following Bloom Taxonomy levels:
Format:\
Each question should have 4 options (A, B, C, D), clear and concise, only one should be correct”””

3.3.2 Coding Challenge Generation

This coding challenge generator harnesses the power of RAG and LLM technologies to create customized, industry-relevant tasks. By integrating the resume data, the system produces challenges across various difficulty levels, from foundational to advanced. These tasks are carefully designed to evaluate multiple parts of coding proficiency: language syntax mastery (Remember), algorithmic thinking (Think), efficient problem-solving (Apply), and domain-specific concept application (Analyse). The RAG component ensures that challenges are grounded in resume contexts, while the LLM generates diverse, relevant problem statements. This combination results in a comprehensive assessment tool that replicates actual coding interviews. Unlike general coding tests, this system's output is finely tuned to each candidate's background, providing a more accurate assessment of their ability to apply knowledge in practical scenarios.
The prompt which generates the coding challenge:
The coding challenges should focus on practical tasks relevant to what a recruiter might ask a final-year student. The tasks should involve identifying errors, completing code snippets, and writing algorithms of simple to moderate complexity. The challenges should cover a range of topics, from fundamental concepts to medium-level problems, and be clear, concise, and aligned with the candidate's resume skills. Additionally, the challenges should include examples of typical coding questions to effectively evaluate the candidate's abilities.

3.3.3 Interview Question Generation

The interview question generator integrates the Google Gemini-1.5-flash model, an advanced LLM, with a RAG system to create highly personalized interview questions. This approach conducts a thorough analysis of the candidate's resume, extracting key details about their experience and skills. The LLM, using its vast parameter space, generates contextually appropriate questions across various assessment dimensions: technical expertise, problem-solving aptitude, leadership potential, communication proficiency, and cultural fit. The RAG component ensures questions are based on relevant industry contexts and customized to the user’s background. Leveraging the LLM's sophisticated language processing, the system generates customized, open-ended questions and intelligent follow-ups, enabling a dynamic evaluation of critical thinking and reasoning skills. This method surpasses common questioning, offering a customized and comprehensive assessment tool that closely matches expert human interviewers while maintaining consistency and minimizing bias in the interview process.

Prompt for Interview Question Generation:
“““Based on the candidate's resume {analysis} and the identified skills, experience, and education, generate a set of {num_questions} interview questions that assess their fit for the position at our company. The questions should cover topics such as problem-solving abilities, leadership skills, communication skills, cultural fit, etc. Additionally, include follow-up questions to probe deeper into the candidate's responses and evaluate their thought process.
Note:The questions should be specific to the field of {selected_domain}.
Questions should be both knowledge base and industry application level also which include practical application of knowledge but only based on {selected_domain}.
The reference should be taken from the resume analysis but the questions generated should be strictly based on {selected_domain}.”””

IV. Result Analysis

The evaluation framework for the generated summaries utilizes a comprehensive approach that comprises of three key components: Context, Ground Truth, and Generated Summary, to thoroughly assess the quality of the summaries using both Factual Alignment Score and Inclusion Score, The Resumes Dataset is created using various domain specific resumes.

- Context: Detailed information extracted from resumes served as the foundational data for analysis.
- Ground Truth: A carefully curated summary representing the key content of the resumes, used as a benchmark for comparison.
- Generated Summary: The output produced by the model, which was evaluated against the Ground Truth.

The **Factual Alignment Score** was calculated by combining token overlap, synonym matching, and semantic similarity. Token overlap was enhanced using WordNet [23] to expand the Ground Truth tokens with synonyms. Semantic similarity was measured using the paraphrase-MiniLM-L6-v2 model, which calculated cosine similarity between BERT based embeddings of the Ground Truth and the Generated Summary [24].

$$Precision = \frac{|Overlap\ tokens|}{|Generated\ Tokens|}$$

$$Recall = \frac{|Overlap\ tokens|}{|Ground\ Truth\ Tokens|}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The F1-Score**,** derived from the precision and recall of token overlap, was combined with the semantic similarity score, then normalized and weighted to quantify how well the Generated Summary aligned with the factual content of the Ground Truth [25].

$$Factual\ Alignment = \min\left(\frac{F1 + Cosine\ Similarity}{2} \times w1\ ,1\right)$$

The inclusion Score evaluated how well the generated Summary captured the essential elements of the Ground Truth. It calculated the proportion of Ground Truth tokens (including synonyms) present in the summary and incorporated semantic similarity to enhance coverage.

$$Inclusion\ Score = \frac{|Matched\ Ground\ Truth\ tokens \cap Generated\ Tokens|}{|Ground\ Truth\ Tokens|}$$

The final evaluation score was obtained by averaging the Factual Alignment and Inclusion Scores, providing a balanced and rigorous assessment of the summaries' precision and semantic fidelity.

$$Final\ Score = \frac{Factual\ Alignment + Inclusion\ Score}{2}$$

Combining Factual Alignment and Inclusion Scores, a Final score was calculated shown in Figure 3 to assess how accurately the generated summary captured key facts and critical information from the original document. This dual-layered approach, leveraging both lexical precision and semantic analysis, provided a robust framework for evaluating summary quality. Achieving 83.77% accuracy across 10 resumes, the system demonstrated its effectiveness in extracting domain-specific knowledge while preserving the semantic integrity of the content. These results validate the system's ability to generate accurate, domain-relevant summaries, laying a strong foundation for personalized question generation.

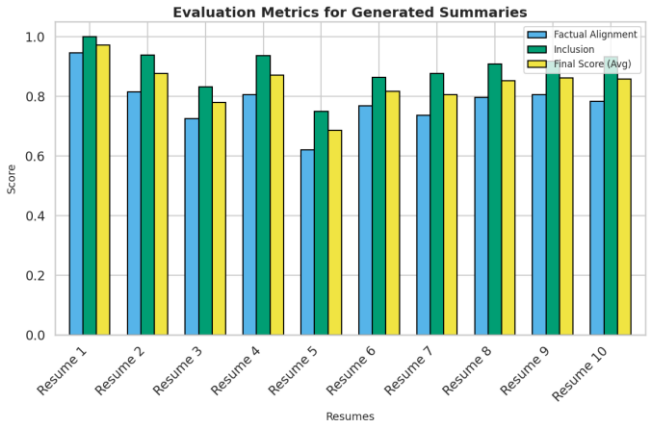

Fig. 3. Evaluation metrics for generated summaries across different resumes

V. Conclusion & Future Scope

This research presents a comprehensive PPA system that makes use of large language models and sophisticated RAG techniques. The PPA system shows notable improvements in personalized interview preparation, domain-specific question generation, and resume analysis. The methodology used in the study, which includes the Generator, Analyzer, and Retriever components, demonstrates a strong framework for processing and evaluating candidate resumes. Efficient and precise information retrieval is ensured by the sophisticated document pre-processing, text extraction, embedding generation, and vector storing methods used by the Retriever component. Through the use of sophisticated language models for thorough resume analysis and domain extraction, the Analyzer component offers a nuanced understanding of candidates' profiles. The Generator component enables personalized placement preparation that generates MCQs, coding challenges & interview questions based on the domains and resumes centrally. The experimental results suggested that the PPA system is able to generate highly relevant and domain-specific questions, showing promise for better placement processes. The flexibility of the system and its applicability across a wide range of that generates different forms of questions and also adapting to varied technical domains. This work advances the field of AI-assisted recruitment and placement preparation by providing a new method that blends domain-specific expertise with LLM. In addition to helping candidates prepare, the PPA system may also help recruiters for designing questions. In the future, the integration of virtual interview avatars will simulate realistic interview scenarios and can be offered iterative feedback on users' performance to improve their interview.

References

[1] A. Di Battista, S. Grayling, E. Hasselaar, T. Leopold, R. Li, M. Rayner, and S. Zahidi, "Future of jobs report 2023," in World Economic Forum, Geneva, Switzerland. https://www. weforum. org/reports/the-future-of-jobs-report-2023, 2023.

[2] L. Mercer, "Global talent trends 2019," Mercer, 2019.

[3] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. K¨uttler, M. Lewis, W.-t. Yih, T. Rockt¨aschel et al., "Retrieval-augmented generation for knowledge-intensive nlp tasks," Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.

[4] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong et al., "A survey of large language models," arXiv preprint arXiv:2303.18223, 2023.

[5] P. Goel and A. Ganatra, "A survey on chatbot: Futuristic conversational agent for user interaction," in 2021 3rd international conference on signal processing and communication (ICPSC). IEEE, 2021, pp. 736– 740.

[6] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: an introduction," Journal of the American Medical Informatics Association, vol. 18, no. 5, pp. 544–551, 2011.

[7] G. Biancini, A. Ferrato, and C. Limongelli, "Multiple-choice question generation using large language models: Methodology and educator insights," in Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization, 2024, pp. 584–590.

[8] H. R. Saeidnia, "Welcome to the gemini era: Google deepmind and the information industry," Library Hi Tech News, no. ahead-of-print, 2023.

[9] E. Malinen, "Interactive document summarizer using llm technology," 2024.

[10] W. Saba, S. Wendelken, and J. Shanahan, "Question-answering based summarization of electronic health records using retrieval augmented generation," arXiv preprint arXiv:2401.01469, 2024.

[11] S. Chataut, T. Do, B. D. S. Gurung, S. Aryal, A. Khanal, C. Lushbough, and E. Gnimpieba, "Comparative study of domain driven terms extraction using large language models," arXiv preprint arXiv:2404.02330, 2024.

[12] P. Varalakshmi and N. M. K. Bugatha, "Ai-powered resume based qa tailoring for success in interviews," in 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS). IEEE, 2024, pp. 1–6.

[13] A. Elmessiry and M. Elmessiry, "Navigating the evolution of artificial intelligence: Towards education-specific retrieval augmented generative ai (es-rag-ai)," in INTED2024 Proceedings. IATED, 2024, pp. 7692–7697.

[14] H. Sajid, J. Kanwal, S. U. R. Bhatti, S. A. Qureshi, A. Basharat, S. Hussain, and K. U. Khan, "Resume parsing framework for e-recruitment," in 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM). IEEE, 2022, pp. 1–8.

[15] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv preprint arXiv:2307.09288, 2023.

[16] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier et al., "Mistral 7b," arXiv preprint arXiv:2310.06825, 2023.

[17] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen et al., "A comprehensive capability analysis of gpt-3 and gpt-3.5 series models," arXiv preprint arXiv:2303.10420, 2023.

[18] A. Hasan, M. A. Ehsan, K. B. Shahnoor, and S. S. Tasneem, "Automatic question & answer generation using generative large language model (llm)," Ph.D. dissertation, Brac University, 2024.

[19] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "Race: Large-scale reading comprehension dataset from examinations," arXiv preprint arXiv:1704.04683, 2017.

[20] B. Koning, "Extracting sections from pdf-formatted cti reports," B.S. thesis, University of Twente, 2022.

[21] V. Ramnarain-Seetohul, V. Bassoo, and Y. Rosunally, "Work-in-progress: computing sentence similarity for short texts using transformer models," in 2022 IEEE Global Engineering Education Conference (EDUCON). IEEE, 2022, pp. 1765–1768.

[22] U. Kumar, P. Kasirajan, G. Sivakamasundari et al., "Smart pdf inquiry hub: A comprehensive solution for efficient pdf document querying and information extraction," in 2024 International Conference on Expert Clouds and Applications (ICOECA). IEEE, 2024, pp. 192–198.

[23] C. Fellbaum, "Wordnet," in Theory and applications of ontology: computer applications. Springer, 2010, pp. 231–243.

[24] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, "Publicly available clinical bert embeddings," arXiv preprint arXiv:1904.03323, 2019.

[25] R. Yacouby and D. Axman, "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models," in Proceedings of the first workshop on evaluation and comparison of NLP systems, 2020, pp. 79–91.