Journal of Holistic and Scientific Studies

Article journey
Received
02/01/2026
Revised
26/01/2026
Accepted
02/02/2026
Online first
9/02/2026

Resume Parsing To Job Listing: A Holistic Ai Framework For Placement Readiness

***Chandrika C*[1*], Dr. Monika Puttaramaiah[2], Vinit Jain[3], *Akshaya M V*[4], *Dr. Gowrishanka*[5]**

[1*]*Department of Machine Learning, BMS College of Engineering, Bangalore, India chandrikac.ai21@bmsce.ac.in*
[2]*Associate Professor, Department of Machine Learning, BMS College of Engineering, Bangalore, India, monika.mel@bmsce.ac.in*
[3]*Department of Machine Learning, BMS College of Engineering, Bangalore, India, vinitjain.ai21@bmsce.ac.in*
[4]*Department of Machine Learning, BMS College of Engineering, Bangalore, India, akshayamv.ai21@bmsce.ac.in*
[5]*Professor, Department of Machine Learning, BMS College of Engineering Bangalore, India, gowrishankar.cse@bmsce.ac.in*

chandrikac.ai21@bmsce.ac.in

***Abstract*— The increase in recession globally, demands a need for proper guidance for the start of the career with job-ready tools. The main factors that hinder students' performance in the present job market are failing to find the right resources, the right jobs that fit their skills, and the right platform to practice for skill evaluation tests in one place. To enable this, we have introduced a novel approach called Job Genie. The approach used in the paper can provide users with a job recommendation engine for job search suitable for their resume, a conversational interview chatbot for boosting confidence in individuals, a resume confidence score provider for a given job description with the correct job title predictor for the uploaded resume, and an aptitude test integrated with LLM for generating new questions with different levels of difficulty and topics, aiding individuals to solve each topic with confidence. Additionally, a course recommendation engine is provided, which can offer users an opportunity to explore courses based on job titles, with beginner to advanced-level course recommendations. The proposed system showed promising results in improving users' confidence, interview readiness, job alignment, and skill development through integrated and personalized support.**

Keywords: Job recommendation engine, Resume analysis, Interview chatbot, Job title prediction, Aptitude test generation, Course recommendation, Career guidance, Skill evaluation.

1. Introduction

Today, the competition in the job market is intense, and it is not only about academic achievements but also includes hard skills, industry awareness, and self-confidence. Job seekers, especially fresh graduates, always find themselves facing hurdles like too many preparation materials, poor guidance, and the inability to decide what the employer needs. Such problems tend to slow the job search and bring anxiety and confusion to applicants' minds.

To fill this gap, Job Genie uses Artificial Intelligence technology to create a complete platform for streamlining and optimizing job preparation. It differs from traditional job portals and interview preparation software, which are entirely stand-alone, by integrating numerous AI modules offering job feed recommendations, real-time aptitude tests, AI-based mock interviews, and career pathway creation.

The new-age technology includes Natural Language Processing, Collaborative Filtering, Large Language Models; with the help of which the platform gives candidates targeted, data-backed suggestions to increase employability.

The current state of the job preparation ecosystem is quite fragmented. Typical job portals like LinkedIn and Naukri tend to list jobs but do not match them with any personalized recommendations based on an individual's skills. Aptitude practice websites deal with stale sample questions that never visit or deviate according to candidates' performance, and simulator mock interviews hardly offer individual feedback. Searching for jobs thereby seems common but does not ensure that a smooth, complete experience will create an inefficiencies-level look across multiple sites.

In contrast, the personalized AI-enabled adaptive platform that Job Genie provides causes the individual's job search and preparation to adapt. For example, the Resume Screening and Job Recommendation module matches candidates according to experience and skill set to potential openings making use of NLP and Collaborative Filtering. The Aptitude Evaluation uses Large Language Models to create a model that generates adaptive questions, provides immediate feedback, and has varied levels of difficulty depending on user performance. Preparing the candidate for interviews, the

AI-driven mock Interview simulates a real-life interview environment using GPT-based conversational AI and provides feedback on communication, technical ability, and behavioral answers. Moreover, Career Roadmap Generator personalizes learning paths with material from sites like Coursera, preparing job hunters for their aspirant jobs.
Job Genie, thus, makes a candidate naturally and completely proficient in minimizing anxiety pertaining to preparations and ultimately maximizing confidence. The paper presents how the system architecture, methodology, and implementation of Job Genie suggest potential evidence that AI-solution-based systems can prove beneficial to transforming the job preparation landscape.

1.1 Literature Review

This section explores various research studies that contribute to advancements in speech processing, job recommendation systems, AI-driven learning models, and voice conversion techniques. The methodologies, findings, and limitations of each study are discussed to provide insights into ongoing developments and future improvements in these fields.

A. Speech-to-Text Translation Using Real-Time Language Conversion

Khandelwal [1] proposed a system to convert one person's speech into text that automatically translates his speech into another user's language. The system uses a combined Automated Speech Recognition and Neural Machine Translation for the purpose of smooth and more rapidly two-way communication across different languages. They stated that the system performs well in their experiments as it was able to process the translations rapidly and the network efficiently. Furthermore, the system can perform well in different scenarios, which ensures that the platform or frequency does not lead to significant changes across mobile devices. The most striking aspect of the system is that even after being trained with a small sample size of one language, the translations are more correct and more rapidly when applied to another. They have already tested the system using actual user conversations, and the system's results have proven to be successfully used. Yet, Several prospects remain and doubts have arisen, especially in terms of the accuracy of speech recognition and how the transcription model would be optimized. Henceforth, the team recommends two strategies to alleviate such concerns the addition of a system pause to the next speech input and the addition of a new language.

B. AI-Generated Medical Research with Credible Referencing

The methodology of Omar et al. [2] analyzed the GPT-4 and Google Gemini's performance in generating medical research by the credibility and factual consistency of the completed assignment. It used the LLM-based text generation tool outputs with the known references and validation to guarantee truthfulness. They determined that GPT-4 provided more sources with true and reliable data, while Gemini's texts were slightly human-like. Progress in the field was, aside from source accuracy, relatively minor, and AI-based medical text remains affected by the enormous vulnerability of hallucination and disinformation. Therefore, the next study on the issue identified should focus on the verification of sources and creating guidelines for using AI in medicine.

C. MCQGen: AI-Based Multiple-Choice Question Generation

MCQGen, developed by Hang et al.[3] based on transformer-based architectures, is an artificial intelligence multiple-choice question generation system oriented on personal learning. The core of any transformer model is its self-attention mechanism, allowing the system to understand the relationship between all words in the input sequence. Further, the authors added knowledge graphs to the model, in which individual nodes are assigned concepts from the content and linked. Graphs not only help select the main content ideas but also ensure generated question alignment. The first step in creating a multiple-choice question is generating a key concept, followed by creating an accurate question, producing the right answer, and generating a reasonable distractor. Besides, MCQGen implements adaptive learning, customized with the user's ability and changing previous user experience. Despite significant progress in the quality of generated questions and user engagement based on such a system compared to traditional rules-based systems, there is still bias in generating questions. Therefore, this structure requires additional training and configuration for each specific field.

D. Learning-Based Job Recommendation System

Deep learning-based job recommendation system introduced by Zhang et al.[4] was designed to improve candidate-job match using representation learning and provided rather disappointing results. While the model was able to convert both job descriptions and resumes into vector embeddings and apply a semantic similarity-based matching, findings demonstrated deep learning to be more personalized than traditional keyword methods. Still, it is important to mention that because of the sensitive dataset's bias, fairness was compromised, and the following enhancements might consider real-time recommendation optimizations.

E. Speech-to-Text Systems Optimized for Real-Time Performance

An AI-powered speech translation system with priorities for low-latency performance was developed by Benita et al. [5]. The methodology includes deep learning ASR and NMT, whereas parallel processing helps to translate with high speed. The experiment showed improvement in accuracy making speech-to-text available even for low-resource languages, a success. Nevertheless, the system is unable to cope with compound sentence structures; therefore, there is room for improvement. Perhaps, embracing hybrid AI models with context in the equation can be the following breakthrough.

F. Optimizing Voice Transmission in Real-Time Applications

To improve the quality of real-time voice transmission, Kumar et al. [6] worked on the already identified problem and provided a possible solution. They sought to reduce the delay and increase the clarity of the audio signal, and also after analysis brought to the academic community a perspective method that allows improving the quality of data transfer. The technique uses noise reduction algorithms, bandwidth optimization, and approaches signal processing from a new side. After the completion of the work, the authors noted that an improvement needs to be developed, which allows you to adjust the system to the characteristics of a particular voice. However, adaptability to diverse voice frequencies remains a concern.

G. AI-Based Job Role Prediction System

Hiremath et al.[7] designed a system that uses a machine learning approach to predict appropriate job roles based on provided resumes. The authors state that the system should be used to help people, especially those who do not fully understand suitable positions for them, and it could also help hiring managers by quickly identifying matching resumes. Their model successfully extracts skills, experiences, and education using NLP and then, using machine learning algorithms such as Random Forests and SVM, predicts appropriate job roles. Furthermore, the authors argue that the newer "predictive" model is way better than the older, rule-based one, largely due to a high degree of accuracy – correctly predicting three out of four possible job roles. However, the model is not quite suitable for predicting job roles immediately after graduation, since this data is not represented in the past. As a result, Hiremath et al. suggest including current labor market data in the further model training.

H. Advances in Voice Conversion: From Statistical Models to Deep Learning

Sisman et al. [8] made an extensive review of the development of voice conversion technologies. They have switched from statistical model methods such as Gaussian Mixture Models and Hidden Markov Models to new methods of deep learning. They also examined techniques based on Generative Adversarial Neural Networks, Variational Autoencoders, and Transformers. The research has shown how the deep learning method has increased the quality of the extracted voice. However, speaker characteristic preservation in voice conversion is still a challenging issue because of the high level of computational resources. In the future, it is beneficial to develop more efficient architectures for real-time voice conversion and consider the use of synthesized speech with respect to ethical standards.

2. Methodology

The proposed methodology for Job Genie involves a job recommendation engine using collaborative filtering and Glassdoor API. The resume analyzer analyses resume to predict suitable job roles using an LSTM-based deep learning model. The ATS confidence module to evaluate matches between resume skills and job description. A conversational chatbot with text-to-speech and speech-to-text functionality and Gemini API to rate answers, and give improved answers and suggestions. Additionally, an aptitude LLM to provide aptitude tests generated using LLM and course recommender to suggest courses from beginner level to advanced level using Coursera API. Each component is described in detail below.

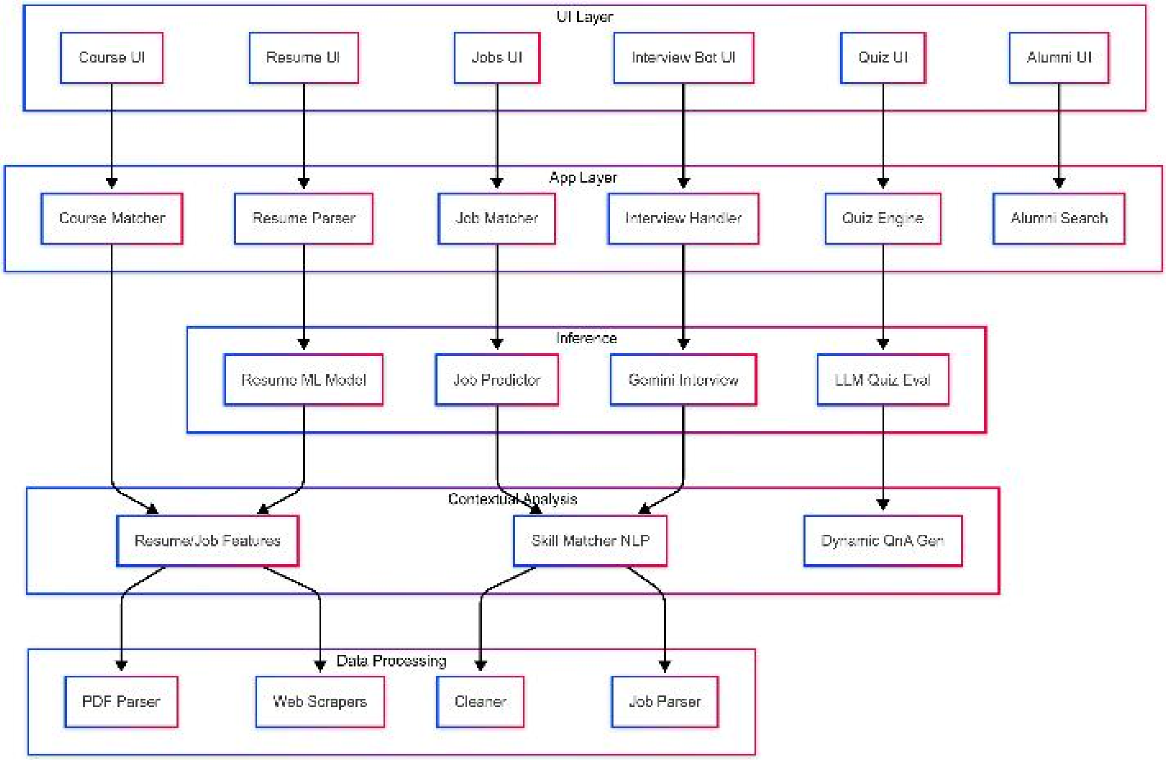


Figure 2. Overall Workflow

2.1 Job Recommendation System

2.1.1 Introduction

The job recommendation system is an application designed to give users job recommendations specifically matching their resume. The user skills and qualifications are scraped, and job requirements from trusted online platforms like Glassdoor and LinkedIn are scraped using natural language processing, and then the two are matched. The main goal of the Job Recommendation System is to cut down the workload involved in job hunting by making job recommendations more relevant and accurate using resume-based matches.

2.1.2 Data Collection

In order to develop a robust job recommendation module, datasets were sourced from open-access job portals and researcher-generated mimicked user resumes. Most of the job listings have been collected in two main ways: through the API and through selenium-based techniques: The API used is called Jooble API. This allowed obtaining job postings from numerous industries and geographic locations in a structured and real-time way. Another method used to collect job-related information was a selenium-based web-scraping tool. Automated scripts were used to scrape publicly available parts of the Glassdoor website in order to collect job postings in compliance with ethical guidelines of web-scraping and the platform's policies. The available job dataset included such characteristics as job titles, descriptions, required skills and qualifications, industry and domain tags, location, and company name. Simultaneously, an anonymized resume corpus in a PDF format was developed. This resume data either open-source or synthetic to represent candidates from various industries.

2.1.3 Workflow

The process starts by a user uploading his resume in PDF format. The file is then uploaded to the system, and the text is extracted using PyPDF2. The spaCy NLP library is subsequently utilized for skill extraction. At the same time, job listings are read from appropriate CSV files automatically generated from Glassdoor and Jooble job platforms systems. The system finally compares the extracted resume skills with job skills and assigns a Confidence score level to the obtained match. The jobs with the Confidence match score greater than the defined constant threshold is then showed to the user. For example, the threshold value can be set at the level of 75%.

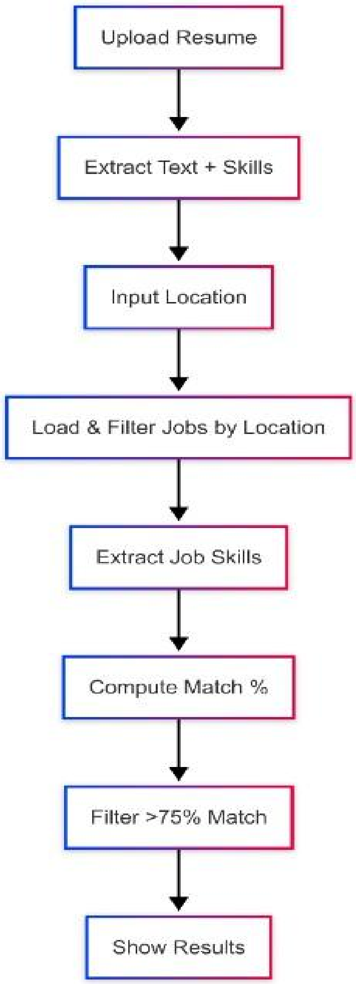


Figure 2.1. Job Recommendation workflow

2.2 Course Recommender Module

2.2.1 Introduction

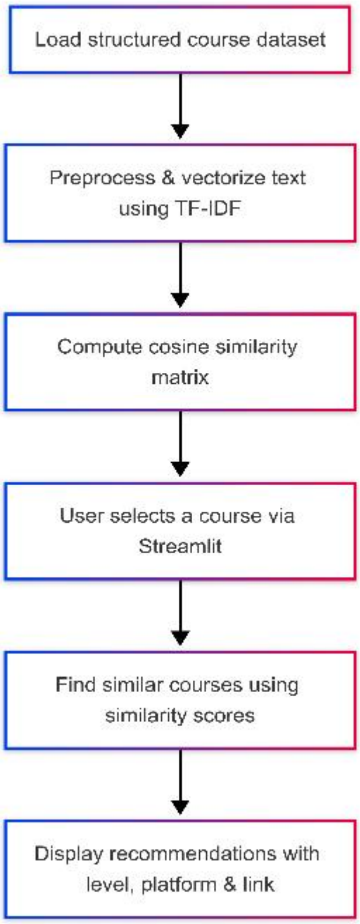
The purpose of the Course Recommender module is to help users discover the online learning opportunities that are most relevant for them. To be more precise, this module recommends the Coursera courses which are most applicable to the interests and goals of the user. The module goes through the dataset of Coursera courses, employs the input preferences of the user, such as what topics they would like to explore and what is the user's general skill level, and recommends the courses that are most similar to the choice of the user. The technique used in the module is content-based filtering with cosine similarity, applied to the textual course names and descriptions. In such a way, the module intends to make the users' experience more focused and efficient, saving their time and making their e-learning journeys more personalized.

2.2.2 Data Collection

The dataset utilized in the Course Recommendation Module was constructed by scraping structured information available on the Coursera website. Data collection was done through automated web scraping technologies based on the Selenium framework to ensure that course metadata could be scraped from publicly available Coursera sections within the platform's ethical scraping policies. The dataset was designed to encompass a wide array of academic and professional development courses from various areas. The following fields were scraped from each course: Course Title, Difficulty Level, Provider / Institution Name, and link of the course. This dataset was used as the base corpus while implementing the recommendation engine. It was cleaned and structured in a Pandas DataFrame to ensure that the data store could be processed sufficiently. This implementation excluded third-party platforms to guarantee that the dataset was consistent and of high relevance. Given the diverse nature of course metadata available on Coursera, it was possible to ensure high quality of course recommendations across various user interests.

2.2.3 Workflow

The workflow behind the Course Recommender system is simple and comprises several essential components. To start, the structured dataset is loaded into a Pandas DataFrame. The dataset is characterized by columns that contain course titles, difficulty levels, platforms and universities that host the course, and detailed descriptions. Then, the text is used to extract meaningful features that can make the course-similarity matrix. More specifically, the course titles and short-level texts are converted into numerical vectors using text-to-matrix methods like TF-IDF. This approach makes it possible to compare the courses based on their content. The cosine is calculated between the vectorized courses, leading to a similarity matrix that will be used to retrieve the most related courses. On the end-user's side, the Course Recommender has a Streamlit interface that allows a user to choose the course they are interested in. This course will then be used as input to retrieve a list of similarly related coursed. Lastly, for each suggestion borrowed from the other course, the user is provided with details of the course such as level of difficulty, the platform it is hosted, and a clickable hyperlink. These additional data allow the user to make a quick judgment on the next most-appropriate course to undertake.



2.3 Course Recommendation workflow

2.3.1 Conversational Interview Chatbot

2.3.2 Introduction

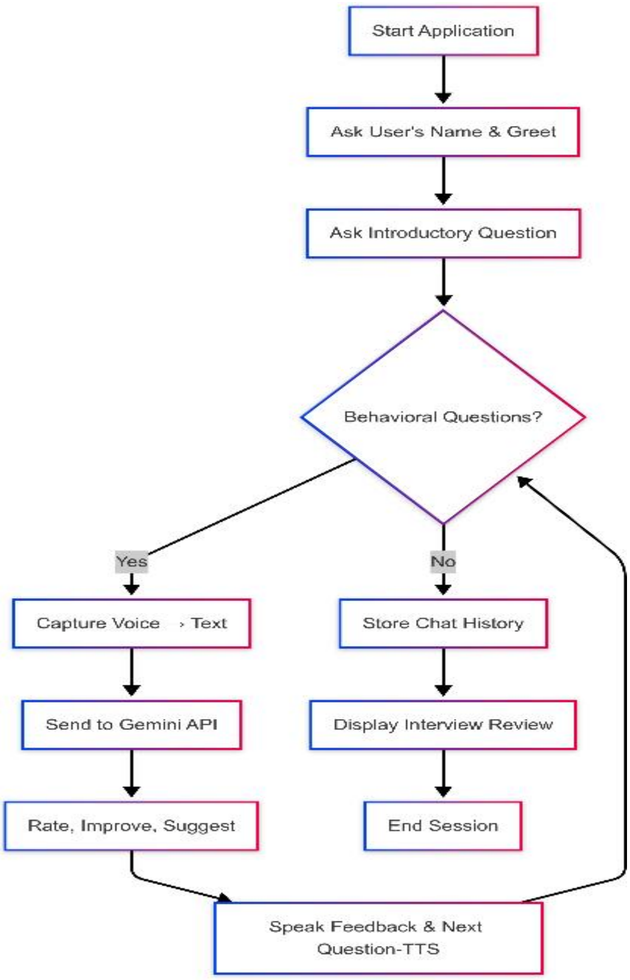
This conversational interview chatbot aids students or individuals to boost their confidence while giving an interview. This involves the integration of Gemini API to conduct mock interviews with voice interaction and feedback.

2.3.3 Data Collection and preparation

Survey forms were circulated among the final-year students from different branches who attended interviews and those who got placed. Mainly questions that focused on behavioral and situational aspects were collected. About 50 questions were collected from trusted sources that were based on soft skills like teamwork, leadership, problem-solving, conflict resolution, adaptability, and communication skills. 40 questions were curated from our college alumni network. All the repeated questions were removed. Each question was validated for clarity and appropriateness for beginner- to intermediate-level candidates. Out of these, 50 questions were finalized for the model to ask the user.

2.3.4 Workflow

The application starts by asking the user their name. Upon entering the name, the user is greeted and questions are dynamically asked with an introduction question at the start followed by 7 more behavioural questions. The voice output is generated using the pyttsx3 library. The assistant asks the questions loudly using TTS(Text-to-speech), and the user responds verbally. Each verbal response is transcribed into text and analysed by Gemini API. This rate the answer provides suggestions, improved answers and also reads aloud. The audio of the user is captured by the speech_recognition library. Finally, the chat history is maintained with assistant questions, user answers and ratings, improved answers and suggestions for overall review of their interview session.



2.4 Interview chatbot workflow

Resume Analyzer

2.4.1 Introduction

Students or individuals prepare their resume but are not sure whether it matches the job description or don't know which job role fits the resume. This ambiguity is avoided using a resume analyzer that predicts a suitable job role based on resume content and provides ATS match score against the provided job description.

2.4.2 Data Collection and Preparation

Resumes that were selected for particular job roles were collected from peers and college seniors. The skills were extracted from these resumes for the job roles they were offered. Job descriptions from various companies were collected for different roles. All of these were combined, and about 85 data points were used to create the dataset.

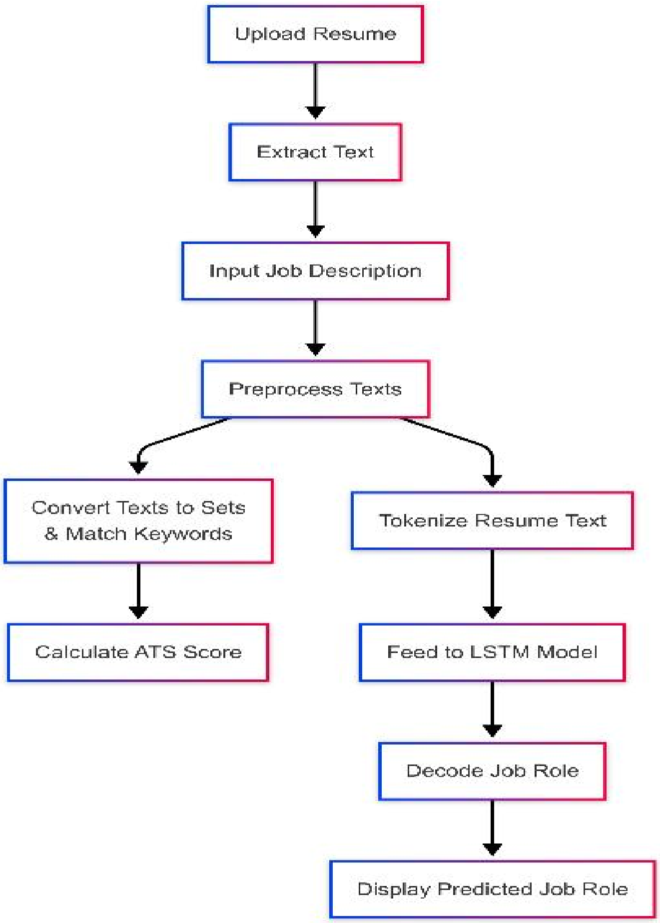
2.4.3 Workflow for ATS Score

The user uploads a resume in PDF format. PyPDF2 library aids in extracting raw text from the resume. Then the user is asked to copy and paste the job description into the textbox. Later both the resume text and job description undergo preprocessing including lowercasing, removing punctuation, tokenizing, removing stop words and lemmatization. The converted text is converted into sets and exact matches are calculated along with percentages using the below formulae:

ATS Score = (Total Job Keywords / Number of Exact Matches) × 100
The final confidence score is displayed.

2.4.4 Workflow for Job predictor

The predictive model is built using a deep learning model, LSTM trained and tested over the dataset of 300 data points with 92% accuracy. The pre-trained model in the .h5 file is used with tokenizer.pkl and label_encoder.pkl for handling tokenization and label encoding. The previously extracted, pre-processed data from the resume is converted to numerical sequences using the pre-trained tokenizer. This is fed to the neural network to predict the job role. The result is decoded to make it readable by inverse transforming the label encoder of the result. The predicted result is displayed.



2.5 Resume Analyzer workflow

Smart Quiz Engine

2.5.1 Introduction

Smart Quiz Engine is a generator-based AI-powered assessment module that provides algorithmically generated multiple-choice question quizzes based on the user's selected topic and complexity level. The solution uses Google's Gemini large language model to generate topic-based questions as well as appropriate answer choices and explanations, producing a personalized, real-time assessment of the user's knowledge level. The interface is realized with an interactive GUI, and LangGraph is used to handle stateful interactions and respond-based adaptive learning bundles.

2.5.2 Data Collection

A custom dataset was needed at the first stage to enable an adaptive and personalized quiz experience, derived from the large language models. A set of various quiz questions was generated using the Gemini API or the appropriate LLM by providing structured prompts, specifically on the topic and sub-topic, question difficulty level, overall number of questions, and a preferable question type regarding the prompt structure. The large batch of questions was generated covering question from multiple domains, including, for example, Python, Data Structures, Operating Systems, and others. The generated text was textually analyzed, and the structured question components – question text, options, verifiable answers, and the answer rationale were programmatically obtained using simple regular expressions to parse the large volume of automatically generated raw text. The next step was cleaning and validating the raw dataset, which included filtering out duplicates and otherwise grammatically correct questions, verifying the topic alignment, and a brief manual check for the logical correctness of verifiable answers. Once everything was put in place, a finalized and validated dataset was used at multiple points to perform training and evaluation, controlled with a quiz generation logic and a user experience behavior in respective quiz beta environments.

2.5.3 Workflow

The system consists of several essential components that combine to deliver a seamless and intelligent quiz experience. The system is linked to Google's Generative AI API, the Gemini 1.5 Pro that produces high-quality quiz content tailored to the topic selected by the user. A Prompt Engineering Layer is used to generate properly structured, clear and consistent prompts based on the user input to ensure that the questions are well-organized and relevant, which results in the AI model giving accurate responses in the required format. The resulting answer is then processed by the Text Parsing Engine that uses regular expressions to very carefully extract the question, answer choices, accurate answer, and explanation to ensure that all of this information is separated and ready for display. Lastly, to make the learned experience truly dynamic and individualized, the system is built using a LangGraph State Machine that models every interaction by a user as a sequence of logical steps, which allows the quiz to change the difficulty level on the fly, provide feedback, and track the prediction to provide a fully online prediction that was less difficult.

2.5.4 LangGraph-Driven Intelligence

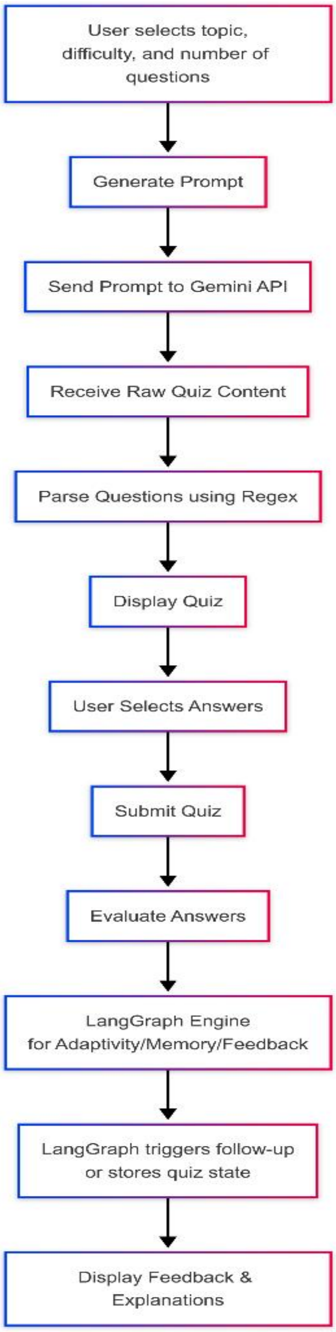
LangGraph manages the Smart Quiz Engine's intelligent flow by running a state-aware, multi-agent system. It essentially works to update the quiz context in real time and monitor the user's progress to seamlessly lead them down the path that will most benefit their learning. Each "agent" component of the LangGraph system is designed to perform specific associated tasks like creating a question, judging the answer, providing the best follow-up question, or updating the user's feedback memory. LangGraph, more explicitly, contributes the following core functions to the system:

Multi-turn Dialogue: Depending on the nature of the response given to the system, hints or follow-up answers may be provided. LangGraph will transfer control from one specialized agent – like the hint agent to the explanation agent – to create a more natural conversational flow.

Memory Management: A given dedicated memory piece will record and interpret users' individual quiz history, such as trend accuracy or performance. LangGraph's feedback helps to tweak the problem's complexity and question design pattern in future quizzes.

Tutor-like Feedback: Instead of merely rating a response right or wrong, the Wise Engine will offer auxiliary information or related knowledge, a problem reformulation, or other visual explanations. LangGraph will, for example, summon an affiliated advice node.

Adaptive Decision-Making: In response to the consumer's output, the Wise Engine will determine the difficulty of the problem: simple, medium, or difficult. LangGraph will also decide the next participated agent who designed the next problem based on the desired difficulty level.



Smart Quiz Engine workflow

3. RESULT

3.1 Job Recommendation Engine

Functionality: It recommends jobs by matching extracted resume text with real job postings.
Evaluation metric: Relevance Score (88%)
Skills were derived from resumes and compared to job descriptions using NLP (via spaCy) and the Jooble and Glassdoor APIs. The Confidence Match Score was calculated as:
Match Score = (Matched Skills / Total Required Skills) × 100
Jobs above a 75% cut-off were presented to the user. The relevance score is the average match percentage across multiple user interactions.
Result: 88% relevance, 80-95% match score above threshold.

3.2 Course Recommendation engine

Functionality: It suggests courses based on user needs scraped from different online platforms.

Evaluation Metric: Relevance Score (88%)
This module used TF-IDF vectorization on course titles and descriptions, and cosine similarity to compare user input with available Coursera courses. The average similarity score of the top suggestions was used to determine the relevance percentage.
Result: 88% relevance

3.3 Resume Analyzer
Functionality: It predicts suitable job title for resume and matches resume with given job description.
Evaluation metric: We used an LSTM-based deep learning model and trained it on a set of 300 labeled resume-job role pairs. The model was tested with a train-test split (usually 80:20), and the accuracy was measured as:
Accuracy = (Correct Predictions / Total Predictions) × 100
ATS Match Score
From a user-uploaded resume and pasted job description, exact keyword matches were tabulated following preprocessing (lowercasing, lemmatization, etc.). The ATS score was determined as:
ATS Score = (Number of Exact Matches / Total Job Keywords) × 100
Result: 92% accuracy, 90% for well optimized resumes.

3.4 Conversational Interview Chatbot
Functionality: It conducts mock interviews and gives rating, improved answer and suggestions.
Evaluation Metric: Average Feedback Score (8.4/10)
Users responded to a Gemini API-driven chatbot that mimicked behavioral interviews. The API scored every response for clarity, organization, and confidence. A session-average score was calculated for each:
Feedback Score = (∑ Individual Answer Ratings) / (Number of Questions)
Result: 8.4(based on mock user testing)

3.5 Smart Quiz Engine
Functionality: It generates topic-wise aptitude questions, evaluates the user answers and gives explanations.
Evaluation Metric: Question Relevance (95%)
Questions were derived from Google's Gemini LLM by topic and difficulty.Relevance was manually and programmatically verified to ensure consistency with the selected topics.
Result: 95% relevance, 90% unique questions generated.

4. Discussion

Numerous platforms serve different purposes like LinkedIn, Glassdoor for job search, Resumenow for ATS score etc. Integrating these platforms can provide individuals with ease in finding all resources and job tools in one place. This is the purpose Job Genie serves.

A. Future Work and Enhancements

Job Genie is a well-balanced tool for job readiness. However, there is continued scope for improvement. One suggestion is the incorporation of a deep learning approach to resume parsing for accurate skill extraction and job suggestions. The addition of sentiment analysis into the simulated interview feedback would also help gauge candidate confidence levels and provide recommendations based on emotional assessment. Also, there is one more area that can be worked on concerning multi-language support, which will help increase accessibility for many job applicants, notably those non-speakers of English. Moreover, integrating blockchain credential validation would help validate candidate credentials and boost employer assurance. Future iterations of Job Genie could also apply reinforcement learning to continually adapt the interview simulation in a way that furthers the goals of simulating reality and the specific role for which the user is preparing. Armed with these advancements, Job Genie is capable of being on the path toward becoming an intelligent and data-driven resource for career preparedness, assisting job-seekers with efficient, inclusive, and impactful intervention.

5. CONCLUSION

The contestants have Job Genie, this great platform that will do magic on their initial preparations for their work workspaces. It is a revolutionary solution powered by artificial intelligence. Such specializations in the existing configurations are those of: natural language processing, collaborative filtering, large language models, as well as GPT-based interview simulations. All these make Job Genie a very personalized, adaptive, and the most effective solution for a job candidate to prepare for a job. Unlike the traditional, fragmented, and non-personalized process of job preparation, Job Genie integrates several critical modules for the candidate's complete career development, such as resume screening, dynamic aptitude testing, AI-driven mock interviews, and a career roadmap.

The paper demonstrates how AI can bridge the divide between candidate capabilities and industry requirements. Automation of key areas of job readiness gives candidates more confidence and can lead to stronger interview performances and greater heritability due to data-driven feedback being provided in real-time. With organizations continuously looking for the crème de la crème in terms of talent professionals, an AI-enabled platform like Job Genie could truly be the game-changer in readying candidates towards disciplined, customized, and focused training. The observations further state that such a system could undoubtedly improve placement success rates tremendously and reduce inefficiencies in job searches, hence being an efficient future labor force enabler.

6. References

1. S. Pokhrel, S. Ganesan, T. Akther, and L. S. Karunarathne Mapa Senavige, "Building Customized Chatbots for Document Summarization and Question Answering using Large Language Models using a Framework with OpenAI, Lang Chain, and Streamlit," *Journal of Information Technology and Digital World*, vol. 6, no. 1, pp. 70-86, 2024. DOI: 10.36548/jitdw.2024.1.006.
2. F. Sufi, "Generative Pre-Trained Transformer (GPT) in Research: A Systematic Review on Data Augmentation," *School of Public Health and Preventive Medicine*, Monash University, Melbourne, VIC, Australia, 2024.
3. Y. Chang, R. Zhang, and J. Pu, "I-WAS: A Data Augmentation Method with GPT-2 for Simile Detection," *Lecture Notes in Computer Science*, vol. 14189, Springer, Cham, 2023. Available: https://arxiv.org/abs/2308.04109.
4. C. Casula, S. Tonelli, and F. B. Kessler, "Generation-Based Data Augmentation for Offensive Language Detection: Is It Worth It?," in *Proc. 17th Conf. European Chapter of the Association for Computational Linguistics (EACL)*, Dubrovnik, Croatia, May 2023, pp. 3359-3377. Available: https://aclanthology.org/2023.eacl-main.244.
5. W. Wu, C. Wang, D. Shen, C. Qin, L. Chen, and H. Xiong, "AFDGCF: Adaptive Feature De-correlation Graph Collaborative Filtering for Recommendations," *arXiv preprint*, arXiv:2403.17416, 2024. Available: https://arxiv.org/abs/2403.17416.
6. A. Casheekar, A. Lahiri, K. Rath, and K. S. Prabhakar, "A Contemporary Review on Chatbots, AI-powered Virtual Conversational Agents, ChatGPT: Applications, Open Challenges, and Future Research Directions," *Computer Science Review*, vol. 52, no. 1, 2024, Art. no. 100632. DOI: 10.1016/j.cosrev.2024.100632.
7. H. Hassani and E. S. Silva, "The Role of ChatGPT in Data Science: How AI-Assisted Conversational Interfaces Are Revolutionizing the Field," *Big Data and Cognitive Computing*, vol. 7, no. 2, Art. no. 62, 2023. DOI: 10.3390/bdcc7020062.
8. Y. Du, D. Luo, R. Yan, H. Liu, Y. Song, H. Zhu, and J. Zhang, "Enhancing Job Recommendation through LLM-based Generative Adversarial Networks," *arXiv preprint*, arXiv:2307.10747, 2023. Available: https://arxiv.org/abs/2307.10747.
9. C. de Ruijt and S. Bhulai, "Job Recommender Systems: A Review," *Vrije Universiteit Amsterdam*, 2021. Available: https://www.researchgate.net/publication/356601605_Job_Recommender_Systems_A_Review.
10. T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, ... and D. Amodei, "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 1877-1901, 2020. Available: https://arxiv.org/abs/2005.14165.
11. K. Vijay, R. Radhakrishnan, and A. Shrivastava, "A Deep Learning Framework for Context-Aware Job Recommendation Systems," *Journal of Computational Intelligence and Applications*, vol. 21, no. 3, pp. 135-150, 2023. DOI: 10.1080/13537760.2023.2224056.
12. S. Lee, Y. Kim, and J. Choi, "Evaluating Candidate Profiles with Natural Language Processing: Enhancing Resume Screening Systems," *Journal of Applied Computing Research*, vol. 58, no. 2, pp. 89-106, 2023. DOI: 10.1016/j.jacr.2023.06.102.