

**Enhancing Career Guidance for Students through Retrieval Augmented Generation in Assistive Software**

|  |  |
| --- | --- |
| **ROLL NO** | **Name** |
| CB.EN.U4CSE21304 | Akhil Swarop S |
| CB.EN.U4CSE21312 | Ganeshkaran M |
| CB.EN.U4CSE21317 | Hanish K R |
| CB.EN.U4CSE21320 | Hidesh Balaji C U |

Name of the Guide : **Bindu K. R.**

Abstract

Students struggle to choose suitable careers due to overwhelming and fragmented information. Traditional guidance is often generic, missing personalized insights. We need an intelligent system that integrates real-time academic and job market data to provide tailored, data-driven career recommendations, enhancing students’ decision-making with minimal user input.

**Introduction**

Navigating today’s dynamic job market can be daunting. With countless job postings and new roles emerging every day, finding opportunities that truly match your unique skills, interests, and career goals often feels overwhelming. Many job seekers spend endless hours sorting through listings, only to end up frustrated by the lack of alignment with their aspirations.

To address this challenge, we’ve developed an innovative job recommendation tool powered by Large Language Models (LLMs) and Retrieval Augmented Generation (RAG). This state-of-the-art system harnesses advanced natural language processing to deliver personalized job recommendations that are precisely tailored to your preferences and expertise.

What sets our tool apart is its ability to adapt to your needs. By allowing you to refine your search with specific skills and interests, it ensures that the recommendations are not only relevant but also aligned with your long-term career goals. This makes the job search more efficient, helping you discover roles that truly resonate with your strengths and ambitions**.**

* 1. Literature Survey and Research Gap Identified

**Paper 1**

**Ref # &Year :** 4 April 2024

**Title :** REST: Retrieval-Based Speculative Decoding

**Problem Statement :** The primary challenge addressed by REST is the inefficiency of traditional language model generation methods, particularly the time-consuming nature of autoregressive decoding. This method requires multiple forward passes of the language model, which can be slow and resource-intensive, especially for large models. REST aims to improve the speed of text generation by utilizing a retrieval-based approach to generate draft tokens, thereby reducing the computational overhead associated with generating each token sequentially

**Dataset :**

* **CodeLlama**:
* A datastore was created using a portion of the Python pretraining code from The Stack, comprising approximately 2.7 million Python code samples, resulting in a datastore size of 27GB.
* **Vicuna**: A datastore was constructed from data derived from Ultra-Chat, which consists of around 774,000 conversations from ChatGPT, yielding a datastore size of 12GB

**Methodology :** Datastore Construction: A datastore is built from relevant datasets, allowing for the retrieval of contextually appropriate tokens.

Token Retrieval: During inference, the input context is used to query the datastore for matching documents. A Trie structure is constructed from the retrieved documents to select the most probable draft tokens.

**Performance Metrics :**

* The paper reports significant speed improvements, achieving a speedup of 1.62X to 2.36X on code or text generation tasks when benchmarked on 7B and 13B language models in a single-batch setting.
* The Mean Generated Length (M) is also considered as a limiting factor for the potential speedup that REST can achieve.

**Paper 2**

**Ref # &Year:** 26 June 2024

**Title** : SEED: Accelerating Reasoning Tree Construction via Scheduled Speculative Decoding

**Problem Statement :** The paper addresses the limitations of Large Language Models (LLMs) in handling complex reasoning and planning tasks. Traditional methods, such as chain-of-thought prompting, are insufficient for these tasks due to their inability to explore intermediate steps effectively. The authors propose SEED, a novel inference framework designed to optimize runtime speed and GPU memory management during reasoning tree construction, thereby reducing inference latency associated with tree-search-based reasoning methods

**Dataset :**

* **GSM8K**: A dataset containing high-quality gradeschool math word problems that require multi-step reasoning.
* **Creative Writing**: This dataset involves generating coherent passages based on four random input sentences, posing challenges in creativity and planning.
* **Blocksworld**: A dataset used to demonstrate the speedup performance of SEED in solving complex planning problems

**Methodology :**

* **SEED** employs a scheduled speculative execution strategy that integrates parallel drafting with speculative decoding.
* The framework utilizes a draft model to generate multiple reasoning paths, which are then evaluated by a state evaluator to determine their contribution to solving the problem.
* This approach allows for efficient management of multiple iterations for thought generation and state evaluation, significantly reducing inference latency.
* The authors also highlight the scalability of their framework to various LLM suites, demonstrating its versatility in different settings

**Performance Metrics:**

* An average speedup of 1.2x in the base setting and 1.5x in the candidate setting across all datasets.
* In the Creative Writing dataset, a speedup of 1.26x was achieved with a reasoning tree depth of 2.
* The performance difference between SEED and AR was found to be within -1.5, indicating that SEED maintains effective performance while enhancing speed

**Paper 3**

**Ref # &Year:** 30 May 2024

**Title :** SpecDec++: Boosting Speculative Decoding via Adaptive Candidate Lengths

**Problem Statement :**

**Inference Latency**: The primary challenge addressed is the inference latency in large language models, which can hinder their usability in real-time applications. Speculative decoding aims to mitigate this by using a smaller draft model to generate candidate tokens for verification by the larger model.

**Sub-optimal Candidate Length**: Previous methods often relied on simple heuristics to select the candidate length (K), which can lead to inefficiencies and sub-optimal performance in the decoding process

**Dataset :**

* **Alpaca Dataset**: This dataset is used to evaluate the performance of the proposed method, providing a benchmark for comparison against existing techniques.
* **GSM8K and HumanEval Datasets**: These datasets are also utilized to assess the effectiveness of SpecDec++ in various contexts, ensuring a comprehensive evaluation across different tasks

**Methodology :**

* **Markov Decision Process (MDP)**: The authors formulate the selection of candidate length K as an MDP, allowing for a more structured approach to decision-making in speculative decoding.
* **Threshold Policy**: The optimal policy derived from the MDP is a threshold policy, which dictates that speculation should stop when the probability of rejection exceeds a certain threshold. This theoretical foundation guides the adaptive selection of candidate lengths.

**Performance Metrics :**

* **Speedup**: The primary performance metric is the speedup achieved in inference time. SpecDec++ demonstrates significant improvements, achieving a 2.04x speedup on the Alpaca dataset, 2.26x on GSM8K, and 2.23x on HumanEval.
* **Improvement Over Baseline**: The method also shows enhancements in performance metrics, with additional improvements of 7.2%, 9.4%, and 11.1% over baseline speculative decoding methods on the respective datasets

**Paper 4**

**Ref # &Year:** 05 July 2024

**Title :** RAMO: Retrieval-Augmented Generation for Enhancing MOOCs Recommendations

**Problem Statement :** The primary challenge addressed by RAMO is the "cold start" problem in course recommender systems, particularly for new users who struggle to find suitable MOOCs (Massive Open Online Courses) due to the overwhelming number of available options. This issue is compounded by the need for personalized recommendations that align with individual learning preferences and career goals

**Dataset :**

* The system would require access to a diverse range of MOOCs and user interaction data to effectively generate personalized recommendations.
* This data would likely include course descriptions, user profiles, and possibly user feedback on courses taken.

**Methodology :**

* RAMO employs a novel approach that integrates large language models (LLMs) with Retrieval-Augmented Generation (RAG) techniques.
* Understand user queries in a conversational manner.
* Retrieve relevant course information based on contextual understanding.
* Generate tailored course recommendations that cater to the unique needs of each user, thereby enhancing the e-learning experience

**Performance Metrics :**

* **Precision and Recall**: To measure the accuracy of the recommendations.
* **F1 Score**: To balance precision and recall.
* **User Satisfaction**: Often assessed through surveys or feedback mechanisms.
* **Cold Start Effectiveness**: Specifically evaluating how well the system performs for new users compared to established users.

**Paper 5**

**Ref # &Year:** 17 August 2024

**Title :** RAGCHECKER: A Fine-grained Framework for Diagnosing Retrieval-Augmented Generation

**Problem Statement :** The primary challenge addressed is the difficulty in evaluating RAG systems due to their modular nature, which consists of both retrieval and generation components. Existing evaluation metrics are often inadequate, failing to capture the complexities of long-form responses and the interactions between the retriever and generator. The goal is to create a comprehensive evaluation framework that provides detailed insights into the performance of RAG systems, identifying strengths and weaknesses in both modules.

**Dataset :**

The dataset used in the study consists of tuples formatted as ⟨query, documents, ground-truth answer⟩. This dataset is curated from public sources and spans across 10 different domains. It is annotated to facilitate the evaluation of RAG systems by providing a reference for assessing the accuracy and relevance of generated responses.

**Methodology :**

* **Claim Extraction:** A mechanism to break down generated responses and ground-truth answers into individual claims.
* **Entailment Checking:** A method to assess whether claims in the generated response are supported by the retrieved context and ground-truth answers.
* **Metric Design:** The framework introduces various metrics, including:Overall Metrics: To provide a holistic view of system performance, such as precision, recall, and F1 score.

**Performance Metrics :**

**Overall Metrics**: Precision, recall, and F1 score based on claim-level comparisons between generated responses and ground-truth answers.

**Retriever Metrics**:Claim recall (the proportion of relevant claims retrieved).

Context precision (the proportion of relevant chunks retrieved).

Generator

**Paper 6**

**Ref # &Year:** 21 April 2024

**Title :** Evaluating Retrieval Quality in Retrieval-Augmented Generation

**Problem Statement :**

The paper addresses the challenges in evaluating retrieval-augmented generation (RAG) systems, particularly focusing on the retrieval models within these systems. Traditional evaluation methods are computationally expensive and show limited correlation between query-document relevance labels and the downstream performance of RAG systems. This indicates a need for a more effective evaluation approach that can better reflect the performance of retrieval models in practical applications

**Dataset** : TriviaQA , HotpotQA , FEVER , Wizard of Wikipedia

**Methodology :**

* Utilizing each document in the retrieval list individually with a large language model within the RAG system.
* Generating outputs for each document and evaluating these outputs based on the ground truth labels of the downstream tasks.
* Using the downstream performance of each document as its relevance label, which is a shift from traditional methods that rely on static relevance labels .
* Aggregating the results using set-based or ranking metrics to assess overall performance

**Performance Metrics :**

* The results indicate that eRAG achieves a significant improvement in correlation, ranging from 0.168 to 0.494 compared to baseline methods.
* Additionally, eRAG demonstrates substantial computational advantages, improving runtime and reducing GPU memory consumption by up to 50 times compared to end-to-end evaluation methods

**Paper 7**

**Ref # &Year: 08 October 2023**

**Title :** Self-Knowledge Guided Retrieval Augmentation for Large Language Models

**Problem Statement :**

The paper addresses the limitations of large language models (LLMs) in retaining complete and up-to-date knowledge. While LLMs perform well without task-specific fine-tuning, they may struggle with incomplete knowledge and can be negatively impacted by irrelevant external information. The goal is to enhance LLMs' performance in question-answering tasks by effectively integrating their internal knowledge with external resources through a method called Self-Knowledge guided Retrieval augmentation (SKR).

**Dataset :**

* **TemporalQA**: Focuses on temporal reasoning questions.
* **CommonsenseQA:** Contains multiple-choice questions that require commonsense reasoning.
* **StrategyQA:** Involves multi-hop reasoning questions.
* **TabularQA:** Tests reasoning with tabular data extracted from Wikipedia.
* **TruthfulQA:** Assesses the truthfulness of responses across various domains like health and politic

**Methodology :**

* Collecting Self-Knowledge: The model's self-knowledge is gathered by analysing its performance on training questions with and without external information.
* Eliciting Self-Knowledge: The model's ability to recognize its knowledge limitations is assessed through various strategies, including direct prompting and in-context learning.
* Adaptive Retrieval: Based on the elicited self-knowledge, the model can decide when to call for external resources to improve its answers to new questions

**Performance Metrics :**

* The performance of the SKR method is evaluated using standard metrics such as accuracy and exact match scores.
* These metrics help quantify how well the model answers questions correctly compared to the ground truth.

**Paper 8**

**Ref # &Year:** December 2023

**Title :** Uni-Parser: Unified Semantic Parser for Question Answering on Knowledge Base and Database

**Problem Statement :** Uni-Parser addresses the difficulty of converting natural language questions into executable logical forms for question answering on structured data sources like knowledge bases (KB) and databases (DB).Existing Limitations: Traditional semantic parsing methods face challenges due to the exponential growth of logical form candidates and often struggle to generalize to unseen data, making it hard to accurately parse questions into logical forms.

**Dataset :**

EvaluationDatasets:

Uni-Parser is evaluated on several datasets:

* GrailQA
* WebQSP
* Focused on Knowledge Base Question Answering (KBQA).Spider and WikiSQL: Used for Database Question Answering (DBQA).

**Methodology :**

Unified Semantic Parsing Framework: Uni-Parser employs a framework that defines primitives (relations and entities in KB, and table names, column names, and cell values in DB) as essential elements. This approach limits the growth of logical form candidates to a linear rate, avoiding the exponential explosion seen in traditional methods.

Generator: Predicts final logical forms by composing top-ranked primitives with various operations (e.g., select, where, count).Contrastive Primitive Ranker: Prunes the search space, enhancing the generator's ability to generalize.T5 Model: The logical form generator is based on a T5 model, which is trained on the datasets to improve performance.

**Performance Metrics :**

* **Exact Match Accuracy (EM):** Measures the accuracy of logical form programs.
* **Answer Accuracy (F1):** Assesses the correctness of the answers generated.

**Paper 9**

**Ref # &Year:** **20 August 2024**

**Title :** Hierarchical Retrieval-Augmented Generation Model with Rethink for Multi-hop Question Answering

**Problem Statement :**

Multi-hop QA requires complex reasoning by integrating multiple pieces of information to answer intricate questions. Existing systems face challenges such as:

* Outdated information.
* Limitations in context window length.
* Trade-offs between accuracy and quantity of information retrieved .

**Dataset :**

HotPotQA\*

\*2WikiMultiHopQA\*

\*MuSiQue\*

\*Bamboogle\*

**Methodology :**

* **Decomposer:** Breaks down complex questions into sub-questions.
* **Definer:** Clarifies the context and requirements for each sub-question.
* **Retriever:** Utilizes a hierarchical retrieval strategy that combines both sparse and dense retrieval methods. - \*Filter\*: Ensures the relevance and quality of retrieved information.
* **Summarizer:** Integrates the answers from sub-questions to provide a **final response.-** The framework emphasizes the retrieval process, which is crucial for producing high-quality results.
* **It also features a single-candidate retrieval method to address the limitations of multi-candidate retrieval .**

**Performance Metrics :**

* HiRAG outperformed state-of-the-art models on three out of four datasets.
* Notable improvements were observed in the Exact Match (EM) index, particularly in the 2WikiMultiHopQA dataset, where HiRAG achieved over a 12% improvement compared to existing methods.
* The results highlight the effectiveness of the Indexed Wikicorpus and the retrieval component in enhancing QA performance .

**Paper 10**

**Ref # &Year :** 22 June 2024

**Title :** Battling Botpoop using GenAI for Higher Education: A Study of a Retrieval Augmented Generation Chatbot’s Impact on Learning

**Problem Statement :**

The study addresses the issue of "Botpoop," the generation of inaccurate or poor-quality information by GenAI chatbots in educational settings. It aims to improve the learning experience by developing a custom Singlish-speaking GenAI chatbot, Professor Leodar, to reduce "Botpoop" and enhance student learning, engagement, and exam preparedness.

**Dataset :**

The knowledge base for Professor Leodar was built from various course materials, including lecture notes, Jupyter notebooks, domain-specific textbooks, and real-time data updates from the "MS0003: Introduction to Data Science and Artificial Intelligence" course at Nanyang Technological University.

**Methodology :**

* The chatbot leverages Retrieval-Augmented Generation (RAG) to provide contextually relevant responses grounded in course content.
* A mixed-methods approach, combining analytics, surveys, and focus group discussions, was used to evaluate the chatbot's impact on student learning. The chatbot's responses were generated using Anthropic’s Claude 3 model.

**Performance Metrics :**

* User engagement: The number of questions asked and interaction peaks during assessments.
* Student feedback: 97.1% of participants reported positive experiences.
* Learning outcomes: A substantial majority (79.4%) highlighted the chatbot’s role in enhancing their understanding of course content.

**Paper 11**

**Ref # &Year :** 02 July 2024

**Title :** Robust Multi Model RAG Pipeline For Documents Containing Text, Table & Images

**Problem Statement :** The primary issue tackled in this study is the inefficiency of existing Multimodal RAGs (Retrieval Augmented Generation) in generating results from documents that contain both images and texts, especially when there are relationships between these elements. - The study aims to propose a solution that enhances the retrieval and generation of results by effectively incorporating these relationships, which is a gap in current methodologies .

**Dataset :**

* **Short-form-type-QA\***
* **\*Long-form-type-QA\***
* **\*MCQ-type-QA\* (Multiple Choice Questions)**
* **\*True-False-type-QA\***

**Methodology :**

* The proposed methodology involves the development of a new Multimodal RAG.
* It integrates both text and images, focusing on their interrelationship.
* The study compares the performance of this new model against existing Multimodal RAGs using the aforementioned datasets.
* Addionally, the proposed model is tested with two different multimodal large language models (LLMs), specifically Open-AI and Gemini, to assess its adaptability and effectiveness in different scenarios .

**Performance Metrics :.**

* The performance of the proposed Multimodal RAG is evaluated based on its effectiveness in generating accurate and relevant results from the datasets.
* The study emphasizes improvements in the generation of results when compared to existing models, although specific metrics (like accuracy, precision, recall, etc.) are not detailed in the provided context .

**Paper 12**

**Year :** 2023

**Title :** Graphsage based named entity recognition for Malayalam

**Problem Statement :**

The paper addresses the challenge of Named Entity Recognition (NER) for the Malayalam language, a Dravidian language characterized by high morphological complexity and agglutinative structure. The main challenges include:

* Limited annotated datasets and tools for Malayalam.
* Difficulty in leveraging conventional NER techniques (e.g., capitalization rules) due to language-specific characteristics.
* Need for an effective method to encode and process contextual relationships between words in Malayalam sentences.

**Dataset :**  
The dataset used for the study involves Malayalam sentences containing named entities. Specific details include:

* The creation of graphs from sentences using a nearest-word method.
* Word embeddings are generated using the **word2vec model** with additional features such as suffixes, part-of-speech tags, and neighboring context.

**Methodology :**

**Graph Construction**: Each sentence is transformed into a graph where words are nodes, and edges are established between nodes based on a nearest-word approach.

**Word Embeddings**: The Continuous Bag-of-Words (CBOW) method in word2vec is used to generate word embeddings.

**GraphSAGE**: The **Graph Sample and Aggregated (GraphSAGE)** algorithm processes the sentence graphs by aggregating feature information from a node's neighbors to create node embeddings. The aggregation function is mean-based.

**Tag Decoders**: **MLP + Softmax**: Used for multi-class classification to independently predict tags for each word based on context-dependent embeddings.

**BiLSTM-CRF**: Utilized for sequence labeling, leveraging bidirectional long short-term memory networks and Conditional Random Fields for enhanced contextual understanding and tag prediction.

**Performance Metrics**

**Precision**:

The system showed high precision in identifying named entities, demonstrating its ability to minimize false positives.

Precision improved over traditional methods due to the effective representation of context using graph structures.

**Recall**:

The system exhibited strong recall, capturing a larger proportion of actual named entities, even in complex Malayalam sentences.

Graph-based processing allowed the model to retain contextual relationships, reducing missed entities compared to sequence-based models.

**F1-Score**:

The F1-score, which balances precision and recall, indicated superior performance over traditional NER models, particularly those based on rule-based or sequence-only approaches.

The GraphSAGE approach demonstrated robustness in handling morphologically complex and agglutinative language structures.

**Paper 13 :  
Year :** 2023

**Title :** GNN based Trainable Dependency Parser

**Problem Statement :** The paper addresses the challenges of dependency parsing for Hindi, a Morphologically Rich and Free Word Order (MoR-FWO) language. Parsing Hindi sentences poses significant challenges due to:

Rich morphology, complex sentence structures, and free word order.

Limited high-quality annotated treebanks for Indian languages.

Traditional dependency parsing methods struggling to handle complex syntactic relationships.

The paper proposes a Graph Neural Network (GNN)-based dependency parser as a state-of-the-art solution for these challenges.

**Dataset :** The dataset used is the Hindi Universal Dependency (UD) Treebank, derived from the Hindi Dependency Treebank (HDTB) project. Key details:

Annotations: Includes dependency relations, Part-of-Speech (POS) tags, and morphological features based on the Paninian Grammar Framework.

Structure: Provided in CoNLL format, with fields like word form, lemma, POS tags (universal and language-specific), dependency relation labels, and more.

Split: Training, validation, and test sets are split in an 8:1:1 ratio.

**Methodology :**

**Word Embeddings:**

GloVe embeddings of 300 dimensions are generated for each word/character to capture semantic relationships.

**Graph Construction:**

Sentences are represented as graphs where words are nodes, and dependency relationships form the edges.

An adjacency matrix or adjacency list is used to represent the graph.

**Graph Neural Network (GNN):**

The GNN processes the graph structure to encode relationships between words using message-passing mechanisms.

Node embeddings are updated iteratively using information from neighboring nodes.

**Dependency Tree Formation:**

The model predicts tree structures and dependency relations:

* + Node-Level Classification: Attributes like POS tags.
  + Edge-Level Classification: Dependency relations between word pairs.

Two methods for tree formation: Greedy Algorithm and Minimum Spanning Tree (MST).

**Model Training:**

A loss function based on negative log-likelihood is used to optimize predictions for tree structures and edge labels.

**Performance Metrics:**

**Accuracy:**

The GNN-based parser achieved a POS tagging accuracy of 92.77%.

**Unlabeled Attachment Score (UAS):**

Demonstrated a high score, indicating accurate identification of syntactic head-child relationships in Hindi.

**Labeled Attachment Score (LAS):**

Exhibited significant improvement over traditional methods due to the ability of GNNs to capture both local and global dependencies.

**Comparison with Machine Learning Models:**

GNN models outperformed traditional classifiers like Decision Trees and Random Forests (accuracy ranged between 43% to 68% for ML models).

The GNN model benefited from its ability to leverage graph structures for contextual understanding.

* 1. **Research Gap**

Current Retrieval-Augmented Generation (RAG) systems have not been widely explored for career path recommendations.  
**Limitations in Prior Studies:**  
 Focus primarily on educational applications.  
 Depend on static datasets that lack essential contextual details, such as user skills and educational background.  
**Challenge:**  
 Lack of dynamic, personalized data in existing RAG systems.  
 This limitation reduces the effectiveness of RAG in providing accurate career guidance.  
**Future Research Directions:**  
 Investigate how RAG systems can use real-time industry trends, psychometric analysis, and :individual user profiles.  
 Aim to create personalized, data-driven career recommendations.  
**Optimization Considerations:**  
 Efforts are underway to minimize datastore size without compromising the performance of RAG systems, following methods that evolved from traditional decoding to RESTful approaches.

* 1. **Problem Statement**

 To develop and evaluate a Retrieval-Augmented Generation (RAG) system that provides personalized career guidance by integrating user-specific data, including academic history and psychometric profiles, to enhance the accuracy and relevance of career recommendations.

* 1. **Module Description**

**1. User Profile and Data Collection Module**

**Description:**

Collects and manages comprehensive user data, including academic history, research interests, skills, hobbies, and psychometric profiles. This module serves as the foundation for generating personalized recommendations.

**Functions:**

* **Academic Data Collection**: Import transcripts, course records, publications, and certifications.
* **Psychometric Profiling**: Incorporate results from personality tests and cognitive assessments.
* **Interest and Skill Input:** Allow users to specify or update their interests, hobbies, and skills.
* **Data Privacy Management:** Ensure all collected data is stored securely and user consent is obtained.

**Input**: User-provided information (academic records, psychometric assessments, interests, skills).

**Output**: A comprehensive and secure user profile for use in recommendation generation.

**2. Data Preprocessing and Normalization Module**

**Description:**

Processes raw user data and external datasets to prepare them for effective retrieval and analysis. It normalizes and enriches data to align with standardized formats and taxonomies.

**Functions:**

* **Data Cleaning:** Remove inconsistencies and errors in user and external data.
* **Normalization:** Standardize data formats (e.g., unify grading scales, course codes).
* **Enrichment**: Map courses, skills, and interests to standardized classifications and ontologies.
* **Feature Extraction:** Identify key attributes from textual data for better matching.

**Input:** Raw user data, external academic and job market data.

**Output:** Cleaned and enriched data ready for retrieval and recommendation processes.

**3. Knowledge Base Construction and Management Module**

**Description:**

Builds and maintains a dynamic knowledge base that aggregates real-time academic publications, emerging research topics, and job market trends relevant to various fields.

**Functions:**

* **Data Aggregation**: Collect data from academic journals, conference proceedings, preprint servers, job postings, and industry reports.
* **Indexing and Storage:** Organize data for efficient retrieval using indexing and semantic embeddings.
* **Continuous Updating**: Regularly update the knowledge base to include the latest information and trends.

**Input:** External data sources (academic databases, job market APIs, industry reports).

**Output:** An up-to-date, searchable knowledge base of academic and market information.

**4. Semantic Retrieval Module**

**Description:**

Retrieves relevant documents and information from the knowledge base that align with the user's profile, focusing on emerging research areas and career opportunities that match the user's expertise and interests.

**Functions:**

* **Semantic Search:** Use advanced NLP techniques to find documents related to the user's interests and skills.
* **Relevance Ranking:** Prioritize results based on relevance scores and recency.
* **Filtering Options:** Allow customization of search parameters (e.g., exclude certain fields, focus on specific industries).
* **Contextual Matching:** Align retrieval results with the user's psychometric profile to enhance suitability.

**Input:** Enriched user profile, updated knowledge base.

**Output:** A curated list of relevant research topics or career opportunities.

**5. Retrieval-Augmented Generation (RAG) Module**

**Description:**

Integrates retrieved information with generative AI models to create personalized and context-rich recommendations for research topics or career paths.

**Functions:**

* **Contextual Generation:** Use retrieved data as context for generating tailored suggestions.
* **Personalization:** Align recommendations with the user's unique interests, skills, and psychometric traits.
* **Novelty Emphasis:** Highlight emerging and underexplored areas that offer opportunities for significant contributions.
* **Consistency Checks:** Ensure generated recommendations are coherent and relevant.

**Input:** Retrieved information, user profile data.

**Output:** Personalized recommendations for research topics or career paths.

**6. User Interface and Experience Module**

**Description:**

Offers an intuitive and engaging platform for users to interact with the system, ensuring ease of use and accessibility.

**Functions:**

* **User-Friendly Design**: Implement an interface that is easy to navigate and understand.
* **Multi-Platform Support:** Ensure compatibility across devices (desktop, tablet, mobile).
* **Interactive Elements:** Include chatbots or virtual assistants to guide users through the process.
* **Accessibility Features:** Support for users with disabilities (e.g., screen readers, adjustable text sizes).

**Input:** User interactions.

**Output:** A seamless and accessible user experience.

**7. Core Technologies and Methodologies**

**Data Scraping :**

**pdfminer Data Scraping Library**

At the heart of the system is a versatile Python library designed for scraping and collecting resume data. This library enables the extraction of key information from various document formats and sources, providing the raw data required for further analysis and processing.

**Key Features:**

* Supports data scraping from resumes in formats such as PDFs, Word documents, and plain text.
* Uses advanced techniques to extract and categorize resume content into structured formats.
* Handles diverse resume layouts and templates, ensuring comprehensive data collection.

**Use Cases:**

1. Scraping and extracting candidate details for storage in databases.
2. Generating structured datasets for downstream analysis and training AI models.
3. Preprocessing raw resume data for AI-driven matching algorithms.

**Fine-Tuned LLM: Gemma 2, 2B**

Gemma 2, 2B is a state-of-the-art language model fine-tuned specifically to handle resumes and job-related data. Its fine-tuning helps it recognize industry-specific and the unique patterns of resume language. The model outputs structured data in JSON format, making integration with other system components seamless.

**Fine-Tuning Process:**

* **Data Collection:** We compiled extensive datasets of resumes and job descriptions.
* **Preprocessing:** The data was formatted into JSONL files, complete with labeled annotations.
* **Training:** Using the UNSLOTH.AI platform, we optimized the model to improve accuracy and ensure it aligns with domain-specific requirements.

**UNSLOTH.AI**

This fine-tuning platform played a vital role in customizing Gemma 2, 2B.

**Key Features:**

* Supports large-scale dataset preparation.
* Provides tools to adjust training parameters and hyperparameters for optimal performance.
* Includes detailed evaluation metrics to monitor model quality.

**O\*NET Job Descriptions**

The **O\*NET database** is a valuable resource for detailed occupational information, including job requirements, skills, and qualifications. It serves as the foundation for mapping candidate skills to appropriate job roles.

**How We Use O\*NET:**

1. Aligning extracted skills with relevant job categories.
2. Using SOC codes (Standard Occupational Classification) for classification and reporting.

**DeepEval Framework**

The **DeepEval Framework** ensures the quality of outputs generated by the system. It evaluates job recommendations based on relevance, accuracy, and how well they match the candidate’s profile.

**Metrics Used:**

* **BLEU Score:** A measure of how closely generated content matches the ground truth.

**Applications:**

1. Verifying that job recommendations align with the candidate's skills and preferences.
2. Assessing and improving the LLM outputs during development.

**SERP API**

To provide up-to-date job postings, the system integrates with the **SERP API**, which fetches real-time data from search engines.

**Example Workflow:**

1. Searching for job listings on platforms like Google Jobs based on specific skills.
2. Filtering results to fit the candidate’s location, role preferences, and other criteria.

**Phys Embeddings for Semantic Matching**

**Phys embeddings** enhance the system's ability to analyze and match skills to job descriptions. These embeddings are dense vector representations that capture the context and meaning of textual data.

**Advantages:**

* Leverages transformer-based models for high-quality semantic matching.
* Reduces mismatches in job recommendations by understanding subtle nuances in text.

**Applications:**

1. Lower mismatch rates between candidates and job roles.
2. Improved understanding of complex job descriptions, leading to better recommendations.