



**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**

Navigating career choices can be overwhelming for students, who often face a flood of information without clear guidance on what’s truly relevant to them. Traditional career advice tends to be one-size-fits-all, which misses the mark for students needing insights tailored to their unique goals and the changing job market. This paper introduces an AI-driven career guidance system based on Retrieval-Augmented Generation (RAG). By pulling together real-time data from academic sources and the job market, this system delivers personalized career recommendations that require minimal input from students. Our approach aims to simplify decision-making by providing students with relevant, data-informed options that align with their interests and the latest industry trends, helping them make confident, informed career choices.

**SECTION I.**

**Introduction and Motivation**

Identifying specialized and developing subjects is critical in a competitive study field for academics who want to make important and novel contributions. Nonetheless, there is a significant obstacle due to the abundance of previously published works and the steady stream of fresh data. In order to locate certain topics that complement their specialized knowledge and interests, researchers frequently find it difficult to sort through this massive amount of data. When academics are trying to focus on newly emerging research possibilities and have specific areas of knowledge, this difficulty becomes even more apparent.

We provide a software approach based on Retrieval Augmented Generation (RAG) to tackle this problem. This tool helps academics find unique study topics based on their individual queries by utilizing cutting-edge natural language processing and retrieval techniques.

The system offers tailored recommendations by letting users add or remove specific hobbies and talents. By streamlining the research process, lowering information overload, and assisting researchers in concentrating on novel and uncharted territory, this strategy seeks to match their unique interests and strengths with their task. This instrument not only increases the effectiveness of research but also encourages more focused and significant scholarly contributions.

**SECTION II.**

1. **Literature Survey**

**Paper 1:**

**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

**Datasets:**

* **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:**

**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

**Datasets:**

* **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:**

**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

**Datasets:**

* **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:**

**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

**Datasets:**

* 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:**

**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.

**Datasets:**

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:**

**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

**Datasets**: 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:**

**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).

**Datasets:**

* **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.
* The results demonstrate that SKR outperforms both chain-of-thought based and fully retrieval-based methods, indicating its effectiveness in leveraging both internal and external knowledge

**Paper 8:**

**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.

**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:**

**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.

**Datasets:**

* 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 afinalresponse**.** 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:**

**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.

**Datasets:**

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:**

**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.

**Datasets:**

* **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:** 2 March 2015

Title: Semantic Search Using a Similarity Graph

**Problem Statement:**

Develop an advanced semantic search engine capable of retrieving and ranking documents based on semantic similarity to a given query. Unlike traditional keyword-based retrieval systems, this engine should leverage a similarity graph to account for the meaning of words and phrases within documents and queries. The system should handle queries where relevant documents may not share exact keywords with the input query but are semantically aligned, such as retrieving a document on "Ford" and "Chrysler" for a query about "cars."

**Datasets**:

* **Cranfield Benchmark**:
  + The primary dataset used in the experiments is the Cranfield benchmark, which consists of 1400 short documents focused on the physics of aviation. Each document includes a title and a brief body, typically around 10 lines long. This dataset is well-known in the field of information retrieval and provides a solid foundation for testing search algorithms
* **Natural Language Queries**:
  + Alongside the documents, the benchmark includes 225 natural language queries.
* **Relevance Judgments**:
  + The relevance of the documents to each query was determined by experts in the field.

**Methodology**:

* Similarity Graph: The core method introduced in this paper is the use of a similarity graph. This graph represents the degree of semantic similarity between terms, which can be words or phrases.
* Probabilistic Model: The paper utilizes a probabilistic model to rank documents based on their relevance to the input query.
* TF-IDF Enhancement: The authors enhance the traditional Term Frequency-Inverse Document Frequency (TF-IDF) algorithm by integrating the similarity graph.
* Experimental Validation: The proposed algorithm is validated through experiments on the Cranfield benchmark, which includes 1400 documents and 225 natural language queries.
* Natural Language Processing: The method also incorporates natural language processing techniques to interpret user queries and identify relevant documents based on their semantic content.

**Performance Metrics**:

* Mean Average Precision (MAP):
  + The primary metric used to assess the performance of the semantic search algorithm is the Mean Average Precision (MAP) score.
* Relevance Scoring:
  + The documents are assigned relevance scores based on their importance to the queries. Highly relevant documents receive a score of 1, while less relevant documents are scored 2, 3, or 4, indicating decreasing levels of relevance
* Comparison with Apache Lucene:
  + The paper compares the MAP scores of the proposed algorithm with those of the Apache Lucene algorithm, which is based on keyword matching. The results indicate that the semantic search algorithm outperforms Lucene, particularly when considering documents with relevance scores from 1 to 4.
* Precision Calculation:
  + For each query, the precision is defined as the fraction of retrieved documents that are relevant.

**Paper 13:**

**Year: 4** October 2024

**Title**: **Resspar: AI-Driven Resume Parsing and Recruitment System using NLP and Generative AI**

**Problem Statement:**

Develop a web-based resume parsing and recruitment system, "Resspar," that leverages NLP, Generative AI, and prompt engineering to streamline candidate selection processes for recruiters. The system should allow users to upload resumes in PDF format, parse relevant data such as names, emails, phone numbers, and skills using AI-driven algorithms, and store extracted information in a structured SQLite database for efficient retrieval and filtering.

**Datasets**:

* **Parsed Resume Data**: The primary dataset consists of resumes uploaded by users in PDF format.
* **User Authentication Data**: Another critical dataset includes user authentication information, which is securely stored in the SQLite database.
* **Job Role and Skill Criteria**: The system allows users to specify job roles or domains and search for specific skills required for particular positions.
* **Text Index for Resumes and Job Descriptions**: The system generates a text index for each resume and job description, which is crucial for efficiently searching and matching relevant documents based on their content.

**Methodology**:

* **Natural Language Processing (NLP)**: Resspar leverages NLP techniques to analyze and understand the content of resumes.
* **Generative AI**: The system utilizes Generative AI, specifically through the Google GenAI API, to interpret visual information from resume images and generate structured text output.
* **Data Extraction Algorithms**: Advanced algorithms are implemented to automate the extraction of essential information from resumes.
* **SQLite Database Management**: Resspar employs SQLite as its database management system to securely store parsed data.
* **Filtering Mechanism**: The system features a powerful filtering mechanism that allows users to specify job roles or required skills.
* **Security Measures**: Resspar incorporates security measures such as password hashing techniques to protect user authentication data

**Performance Metrics**:

* **Accuracy of Data Extraction**: One of the primary metrics would be the accuracy with which the system extracts relevant information from resumes.
* **Processing Speed**: The time taken to parse a resume and present the extracted data.

1. **Research Gaps**

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.

**SECTION III.**

* 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. 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.

1. **Algorithm:**

## 1. Initialization

### 1.1 Define Knowledge Bases:

1.1.1 KB1: O\*NET Database with Comprehensive occupation titles and detailed descriptions

1.1.2 KB2: LinkedIn Job Scraping Module- Current job postings scraped from the Arbeitnow API based on relevant job titles.

### 1.2 Initialize System Components:

1.2.1 Input Module: Collects user inputs including resumes, psychometric profiles, and academic history.

1.2.2 Keyword Extraction Module: Identifies essential skills and keywords from user data using Natural Language Processing (NLP) techniques.

1.2.3 Retriever Module: Fetches relevant job descriptions from KB1 based on extracted keywords.

1.2.4 Similarity Search Module: Measures semantic similarity between user skills and job descriptions.

1.2.4 Ranking Module: Orders job matches based on relevance scores derived from similarity metrics.

1.2.5 Scraping Module: Retrieves job postings from the Arbeitnow API for top-ranked job titles.

1.2.6 Generator Module: Creates personalized career guidance using aggregated information.

1.2.7 Visualization Module: Displays skill match percentages and skill gap analysis.

1.2.8 Unsupervised Clustering Module: Performs topic modeling on O\*NET occupation descriptions to uncover underlying themes.

1.2.9 Output Module: Presents final recommendations and job listings to the user.

### 1.3 Set Parameters:

1.3.1 Set Number of top relevant jobs to identify

1.3.2 Set Criteria for keyword relevance and match scores.

1.3.3 Set Parameters for LinkedIn searches, such as job titles.

1.3.4 Set LDA paramteters for topic modeling (e.g., number of topics, iterations).

### 1.4 Load Necessary Resources and Models:

1.4.1 Load spaCy For skill extraction.

1.4.2 Load SentenceTransformers for generating semantic embeddings.

1.4.2 Load LDAModel for unsupervised clustering on O\*NET data.

## 2. User Input Collection and Processing

### 2.1 Collect User Inputs:

2.1.1 User uploads a resume in PDF or DOCX format.

2.1.2 User inputs such as personality traits, strengths, and work preferences.

2.1.3 User inputs details including degree, institution, graduation year, and certifications.

2.1.4 User inputs work and location preferences

### 2.2 Extract and Preprocess Information:

2.2.1 Extract text from the uploaded resume.

2.2.2 Identify and extract relevant skills from the resume text.

2.2.3 Consolidate psychometric inputs into a structured format.

2.2.4 Aggregate academic details into a comprehensive summary.

### 2.3 Store Processed User Profile:

2.3.1 Combine extracted data (skills, academic history, psychometric profile) into a structured format for further processing.

## 3. Keyword Extraction and Retrieval

### 3.1 Extract Keywords and Skills:

3.1.1 Utilize the Keyword Extraction Module to identify and extract key skills, competencies, and relevant keywords from the user's resume and profile.

### 3.2 Retrieve Relevant Job Descriptions (Retriever Module):

3.2.1 Extract skills and keywords.

3.2.2 Query KB1 (O\*NET Database) to fetch job descriptions that align with the extracted keywords.

3.2.3 Return a subset of job descriptions from KB1 relevant to the user's profile.

## 4. Similarity Measurement and Ranking

### 4.1 Compute Similarity Scores:

4.1.1 Use the Similarity Search Module to generate semantic embeddings for user skills using the SentenceTransformer model.

4.1.2 Compare these embeddings with precomputed embeddings of job titles from KB1 using cosine similarity.

4.1.3 Return similarity scores indicating how closely each job description matches the user's skills.

### 4.2 Assign Relevance Scores:

4.2.1 Score each job based on the similarity metrics to determine its relevance to the user's profile.

### 4.3 Rank Jobs by Relevance:

4.3.1 Order all jobs in KB1 based on their relevance scores in descending order.

### 4.4 Select Top\_N\_Jobs:

4.4.1 Choose the top N most relevant jobs for further processing.

## 5. Job Scraping from LinkedIn (KB2)

### 5.1 Configure Scraping Parameters:

5.1.1 Set search criteria based on the selected top job titles, including location preferences and job type.

### 5.2 Execute Scraping:

5.2.1 Collect current job postings from LinkedIn by searching the top job titles.

### 5.3 Aggregate Job Postings:

5.3.1 Compile details such as job title, company name, location, job description, application links.

## 6. Generation of Personalized Career Guidance

### 6.1 Prepare Input for Generator Module:

6.1.1 Combine user data (skills, academic history, psychometric profile), top job matches, and scraped job postings.

### 6.2 Invoke Generator Module:

6.2.1 Feed the consolidated input as a prompt to the generator module.

### 6.3 Integrate Job Postings:

6.3.1 Embed relevant job listings with direct application links into the guidance.

## 7. Visualization and Analysis

### 7.1 Skill Match Visualization:

7.1.1 Display skill match percentages for recommended jobs using Plotly bar charts.

### 7.2 Skill Gap Analysis:

7.2.1 Identify and visualize missing skills required for top job matches, suggesting areas for development.

### 7.3 Unsupervised Clustering with LDA:

7.3.1 Apply Latent Dirichlet Allocation (LDA) to uncover topics within O\*NET occupation descriptions.

## 8. Termination and Feedback

### 8.1 Finalize Recommendations:

8.1.1 Present the personalized career guidance and relevant job postings to the user in a structured format.

### 8.2 End of Algorithm:

8.2.1 Conclude the RAG process, ensuring all user recommendations are delivered and stored.

1. **References:**

* **Paper 1** -He, Z., Zhong, Z., Cai, T., Lee, J.D. and He, D., 2023. Rest: Retrieval-based speculative decoding. *arXiv preprint arXiv:2311.08252*.
* **Paper 2** -Wang, Z., Wu, J., Lai, Y., Zhang, C. and Zhou, D., 2024. SEED: Accelerating Reasoning Tree Construction via Scheduled Speculative Decoding. *arXiv preprint arXiv:2406.18200*.
* **Paper 3** - Huang, K., Guo, X. and Wang, M., 2024. SpecDec++: Boosting Speculative Decoding via Adaptive Candidate Lengths. *arXiv preprint arXiv:2405.19715*.
* **Paper 4** - Rao, J. and Lin, J., 2024. RAMO: Retrieval-Augmented Generation for Enhancing MOOCs Recommendations. *arXiv preprint arXiv:2407.04925*.
* **Paper 5-** Ru, D., Qiu, L., Hu, X., Zhang, T., Shi, P., Chang, S., Cheng, J., Wang, C., Sun, S., Li, H. and Zhang, Z., 2024. RAGChecker: A Fine-grained Framework for Diagnosing Retrieval-Augmented Generation. *arXiv preprint arXiv:2408.08067*.
* **Paper 6** -Salemi, A. and Zamani, H., 2024, July. Evaluating retrieval quality in retrieval-augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 2395-2400).
* **Paper 7** - Wang, Y., Li, P., Sun, M. and Liu, Y., 2023. Self-knowledge guided retrieval augmentation for large language models. *arXiv preprint arXiv:2310.05002*.
* **Paper 8** - Liu, Y., Yavuz, S., Meng, R., Radev, D., Xiong, C. and Zhou, Y., 2022. Uni-parser: Unified semantic parser for question answering on knowledge base and database. *arXiv preprint arXiv:2211.05165*.
* **Paper 9** – Zhang, X., Wang, M., Yang, X., Wang, D., Feng, S. and Zhang, Y., 2024. Hierarchical Retrieval-Augmented Generation Model with Rethink for Multi-hop Question Answering. *arXiv preprint arXiv:2408.11875*.
* **Paper 10** –Thway, M., Recatala-Gomez, J., Lim, F.S., Hippalgaonkar, K. and Ng, L.W., 2024. Harnessing GenAI for Higher Education: A Study of a Retrieval Augmented Generation Chatbot's Impact on Human Learning. *arXiv preprint arXiv:2406.07796*
* **Paper 11** - Joshi, P., Gupta, A., Kumar, P. and Sisodia, M., 2024, June. Robust Multi Model RAG Pipeline for Documents Containing Text, Table & Images. In *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)* (pp. 993-999). IEEE.
* **Paper 12** - L. Stanchev, "Semantic search using a similarity graph," Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015), Anaheim, CA, USA, 2015
* **Paper 13** - A. D, K. S, N. E. R, K. K, J. M. S and R. R, Resspar: AI-Driven Resume Parsing and Recruitment System using NLP and Generative AI," 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI), Coimbatore, India, 2024