Intelligent API for Automated Profile Generation and Job Market Insights

Team Members: Omar Khater
Course: CSCE 689 Programming Large Language Models, Fall 2024

Contents

1 Introduction

The goal for this project was to investigate the potential of using Large Language Models (LLMs) on improving the user experience for an existing social media app to connect creative professionals with opportunities. More specifically, we aimed to develop an Application Programming Interface (API) that allows users to generate high-quality, personalized profile elements tailored to their profession and experience. In addition, it was highly desirable to enable using up-to-date keywords that can be later improve their visibility in the job marketing.

LLMs (e.g., GPT based models) are known for its ability to generate coherent and contextually relevant text. Hence, they are ideal candidate for the task at hand. However, one of the key issues with LLMs is their ability to hallucinate, so returning answers grounded in facts becomes a top priority for many applications such as the one at hand. Retrieval Augmented Generation (RAGs) emerges as an effective less expensive way to add better contextual up-to-date information for the language models to improve the relevancy to the user input.

Several design patterns for integrating RAG in LLM-based applications has been proposed. A key component of this project was to investigate the efficacy of such design patterns in the scope of this project. We briefly describe some common techniques and demonstrate the employed approach.

In this project, the API is independently created and evaluated. Then, it has been deployed to the existing app originally inspired this project. **The GitHub for this project is given in this link**

2 Methedology

2.1 Architecture

To equip LLMs with up-to-date information, organizations use Retrieval Augmented Generation (RAG), a technique that fetches data from company data sources and enriches the prompt to provide more relevant and accurate responses. Common architectures are shown in figures 1 and 2. [1]

The **Simple RAG workflow** (Figure 1) follows a straightforward approach. It uses a single API to retrieve similar documents based on a user query and augment the query with the retrieved documents before passing it to the language model for generating responses. This simplicity makes it easy to implement, but it offers limited flexibility and customization.

The **Customized RAG workflow** (Figure 2) introduces additional steps and customizations. Here, the system retrieves documents using a dedicated Retrieve API. These documents are further processed to generate contextual embeddings and augment the user input before being sent to the language model. This workflow offers more control, allowing fine-tuning and optimization for specific organizational needs.

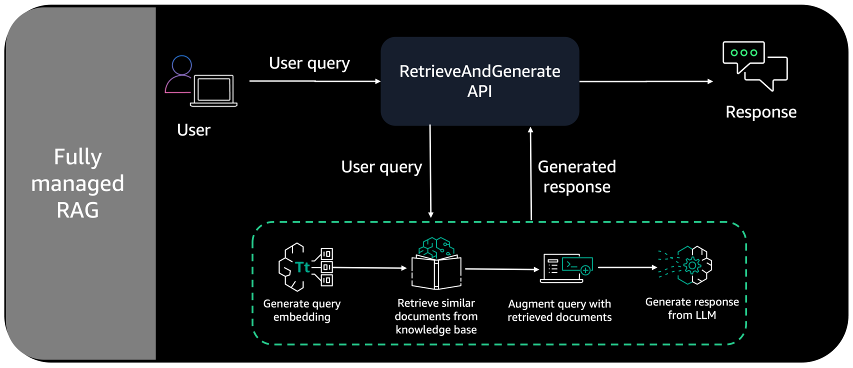


Figure 1: Simple RAG based architecture. [1]

2.2 Data Collection

To support the RAG workflow, a data ingestion pipeline was created (Figure 3). This pipeline begins with the extraction of new data from various sources, chunking the data into smaller components,

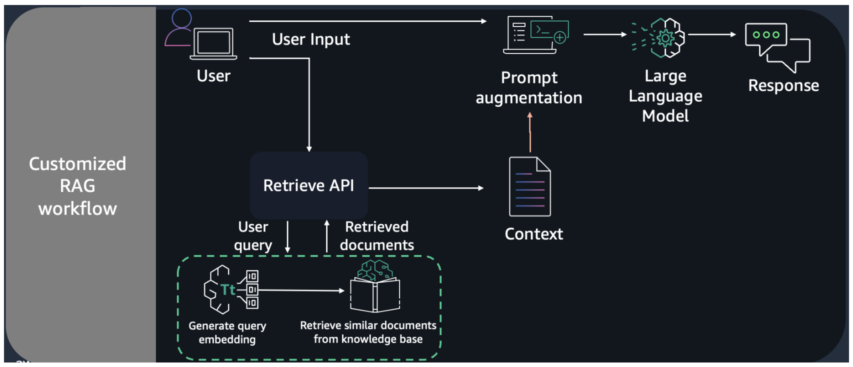


Figure 2: Customized RAG workflow. [1]

generating embeddings using a pre-trained model, and storing the processed data in a vector store for efficient retrieval.

The scripts provided implement this pipeline:

- **Data collection and processing:** The ‘build_job_skills_datasets.py‘ script fetches trending skills for job titles by scraping online sources. The data is organized into a structured dataset, associating each job title with its relevant skills.
- **Database creation:** The ‘build_job_skills_database.py‘ script converts this dataset into a vector database using embeddings generated by the Ollama model. The database allows for similarity-based retrieval by leveraging vector representations of the user input (e.g., profession, experience level).

By combining these steps, the RAG system retrieves contextually relevant information to enhance user queries and generate more accurate responses.

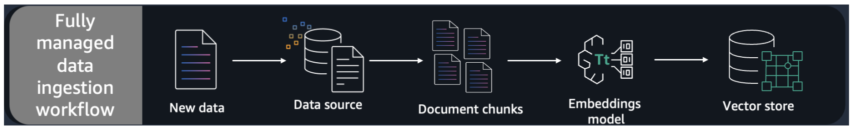


Figure 3: Fully managed data ingestion workflow. [1]

2.3 API Implementation for RAGs Architecture

The RAG architecture leverages a combination of APIs to retrieve, process, and generate content for user queries. The implemented APIs ensure the retrieval of relevant data from vector databases and external sources, followed by generating refined results using large language models (LLMs). The key components of the API implementation are outlined below:

- **Keyword Extraction:** The `fetch_trending_keywords` function utilizes web scraping through the Google Search engine to fetch trending skills for a given profession. It parses the search results using `BeautifulSoup` and extracts a list of relevant keywords. It serves as a fallback in case of unsuccessful retrieval from the RAG.
- **Vector Database Retrieval:** The `retrieve_skills_from_chroma` function interfaces with the ChromaDB vector store. It performs similarity-based searches using embeddings to retrieve keywords relevant to a user-provided profession. The retrieval is threshold-based to ensure only the most relevant data is considered.
- **Profile Generation:** The `generate_profile` function combines user-provided inputs (profession, experience level, and custom keywords) with trending keywords from the vector database or external sources. It uses this enriched input to generate an elevator pitch and project descriptions via GPT-3.5 Turbo.

- **Embedding Integration:** The `get_vectorstore` function initializes the ChromaDB vector database. It employs the Ollama Embeddings model `mxbai-embed-large` to convert text into embeddings and persists the data for efficient retrieval.
- **GPT-Driven Generation:** The `chat_gpt` function interacts with the OpenAI GPT-3.5 Turbo API to generate refined responses. It constructs user-specific prompts and returns structured content suitable for professional profiles.

The implementation workflow is as follows:

1. A user inputs their profession, experience level, and optional keywords.
2. The system checks the ChromaDB vector store for relevant keywords. If no data is found, trending skills are fetched via the Google Search API.
3. The system combines all retrieved keywords and generates a prompt tailored for the LLM.
4. GPT processes the prompt and returns structured outputs, such as elevator pitches and project descriptions.

This implementation ensures efficient retrieval and generation within the RAG framework while maintaining modularity and scalability for future enhancements.

2.4 Prompt Engineering

To ensure optimal performance and effective usage of the LLM, this project emphasized the careful design of prompts. Several techniques such as zero-shot prompting and few-shot prompting were employed depending on the task requirements.

For instance, in generating personalized content like an elevator pitch and a LinkedIn 'About Me' section, we used a structured prompt that dynamically incorporates the user's profession, experience level, keywords, and background information to produce contextually relevant and engaging outputs. The JSON-based response format ensures structured and actionable results.

During the evaluation process, we utilized few-shot learning. This prompt provides clear evaluation criteria and examples to guide the model in assessing the quality of AI-generated content. By leveraging this approach, the model evaluates outputs based on relevance, hallucination, and overall quality, producing structured feedback in JSON format.

In addition to few-shot prompting, we implemented Chain-of-Thoughts reasoning within evaluation prompts to guide the model through a logical assessment process. This technique enhances the accuracy and consistency of the evaluations. While advanced techniques like Tree-of-Thoughts were considered, they were deemed unnecessary for the scope of this project.

2.5 Design Choices

Our API utilize custom relevancy score when building the RAG as shown in equation 1

$$\text{Relevance Score} = \frac{1}{1 + \text{similarity score}} \tag{1}$$

Where the similarity score is calculated by the vector store based on the embeddings of the input user query and the data already stored in the database. We set a threshold parameter that indicates the desired level of similarity between the user input and the retrieved keywords from the RAG. This design enable more control on the output quality from the API and serves as a tuning parameter for further integration with other data sources and custom applications.

3 Evaluation

In figure 4, we demonstrate the overall evaluation procedure followed in this project. Below is detailed description to this procedure.

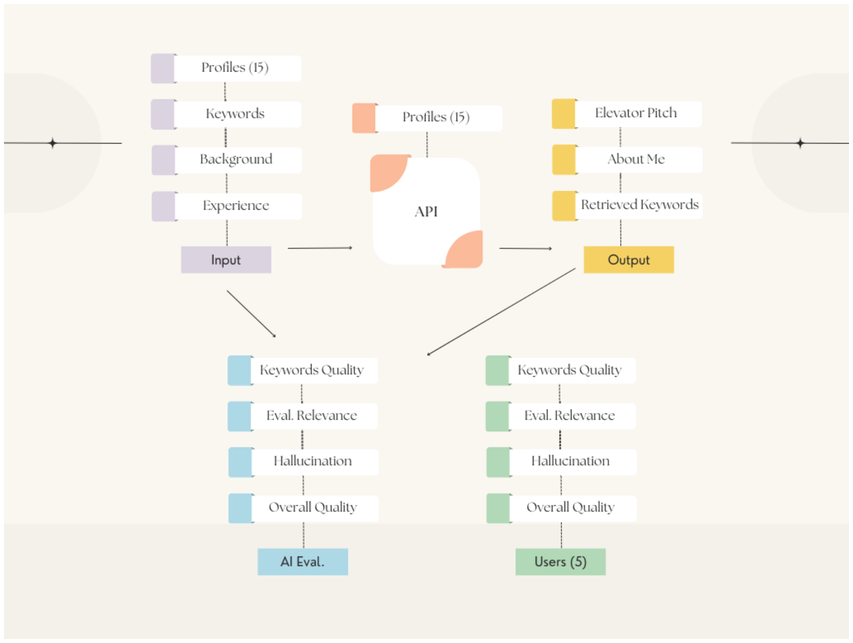


Figure 4: API Evaluation Flowchart.

3.1 AI-Based Evaluation

As described in the project proposal, we aimed to test the API output first on simulated profiles. A GPT-4-based model from OpenAI has been utilized in this context to assess the quality of the generated outputs systematically.

The evaluation process involves the following steps:

- **Input Preparation:** Input profiles were simulated with varying levels of complexity, professions, and keywords. These inputs are saved as JSON files in the designated input directory.
- **Output Generation:** The API generates corresponding outputs based on the inputs, including an elevator pitch, an "About Me" section, retrieved keywords, and additional metadata. These outputs are saved in a structured JSON format in the output directory.
- **Quality Assessment:** The GPT-4 model evaluates each input-output pair based on specific criteria:
  - **Keywords Quality:** Relevance and appropriateness of the keywords used in the output.
  - **Relevance:** How well the generated content aligns with the provided input (profession, background, and keywords).
  - **Hallucination:** The degree to which the output introduces content not implied by the input.
  - **Overall Quality:** A holistic score reflecting coherence, fluency, and utility of the output.

The GPT-4 model provides a structured JSON evaluation for each profile, which is stored in the evaluation directory. These evaluations are then consolidated into a single CSV file for analysis. The consolidated results are saved as `API_evaluation_by_ai.csv`, capturing scores and explanations for each input-output pair.

3.2 Human Evaluation

To complement the AI-based evaluation, human evaluators reviewed the generated profiles to provide qualitative insights into the strengths and weaknesses of the API outputs.

- **Procedure:**
  - Human evaluators were given the input profiles along with their respective API outputs.
  - They assessed the outputs based on same criteria used with AI.
  - Additional comment free response is collected from the users to get more insights.

The consolidated results are saved as `Human_Feedback_API_Evaluation.csv`

3.3 Comparison

To facilitate the comparison, we created another third sheet `Combined_Evaluation_Sheet.csv` to show the average human scores per simulated profile for each output metric vs the AI evaluation.

4 Discussion

Our experimental results demonstrate the efficiency of our design and the robustness of the architecture used for this problem.

The results documented in `Combined_Evaluation_Sheet.xlsx`, highlight both the potential and limitations of using a large language model (LLM) to assist professionals in crafting optimized profiles with market-relevant keywords. We gained valuable insights into the AI's content generation capabilities and feedback from human evaluators.

High-Performing Content with Senior-Level Input

When provided with input from a senior-level product manager, the AI achieved a near-perfect quality score of 100, while human evaluators rated it at 97 and 99. These results suggest that the LLM can effectively handle advanced-level inputs, producing high-quality profiles that meet both algorithmic and human expectations.

Accuracy and Relevance in Specific Backgrounds

For content generated based on a vague background, the AI's hallucination score was 85, and human evaluators rated relevance at 87, indicating some fabrication. These findings suggest that more specific inputs improve the relevance and accuracy of AI-generated content, reducing instances of hallucination.

Challenges in Innovation and Narrative Flow

In a case involving visual storytelling, human evaluators gave the content a keyword quality score of 82, noting a lack of creativity. While the AI excelled at keyword optimization, it struggled to create engaging and original narratives, an area where human evaluators place higher emphasis.

Discrepancy Between AI and Human Evaluation

Overall, the AI tended to score higher than human evaluators. This discrepancy points to different evaluation criteria: the AI focuses on keyword optimization and analytical precision, while humans value narrative elements, readability, and personal context. These complementary strengths suggest that combining AI-driven market alignment with human creativity can produce optimal results.

Implications and Future Directions

These findings demonstrate the potential of LLMs to enhance professional profiles, especially when provided with well-defined inputs. Future work should aim to reduce hallucinations, improve narrative creativity, and integrate human feedback to balance accuracy with qualitative storytelling.

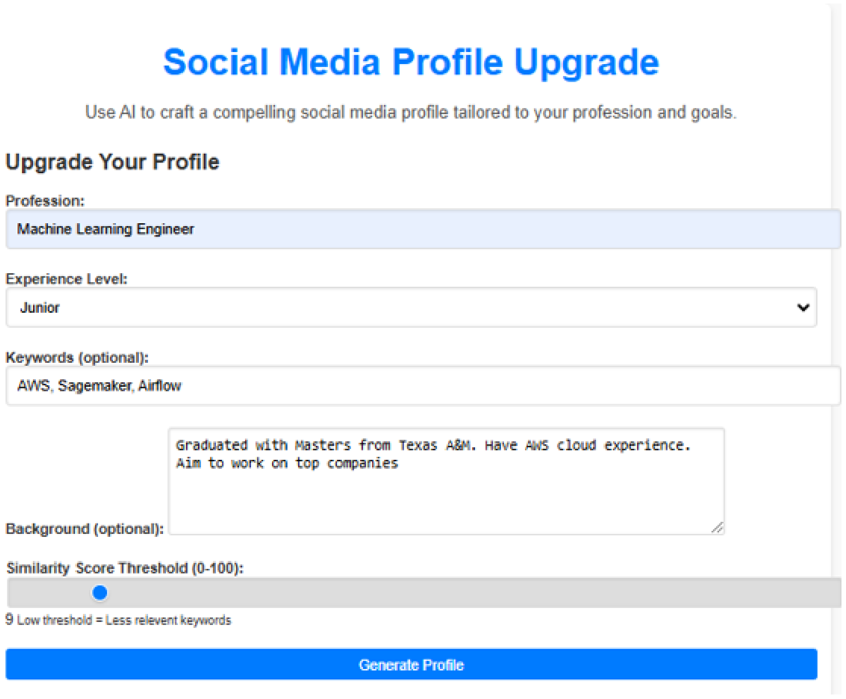
5 Output Delivery

While the user interface (UI) is not of a main focus for this project, we created a simple UI to show how would an application might use this API. In addition, we were able to successfully deploy this feature into the client app despite this is another project out of scope for this project discussion.

5.1 Experimental UI

To visualize possible API integration with other apps, we created simple UI. The UI is a simple flask app that has 2 simple flash cards. The first flash card exemplify the Social Media Profile Upgrade ability of this API where the user is expected to enter some profile elements such as profession, background, some queries, and the experience level. The UI enables generating the desired profile elements. For simplicity, we only show elevator pitch and About Me output text. The keywords retrieved from the RAG can be toggled to have better visibility over the generated content. Also, some concise reasoning about the generated profile process.

Furthermore, the UI enables feedback collection by enabling the users to enter their review on the UI. Finally, the UI enable basic control over the similarity threshold so that the user can control the level of relevancy for the keywords selected to augment generation prompt. A sample screenshot from the user input and the generated output is shown in figures 5 and 6


Social Media Profile Upgrade

Use AI to craft a compelling social media profile tailored to your profession and goals.

Upgrade Your Profile

Profession: Machine Learning Engineer

Experience Level: Junior

Keywords (optional): AWS, Sagemaker, Airflow

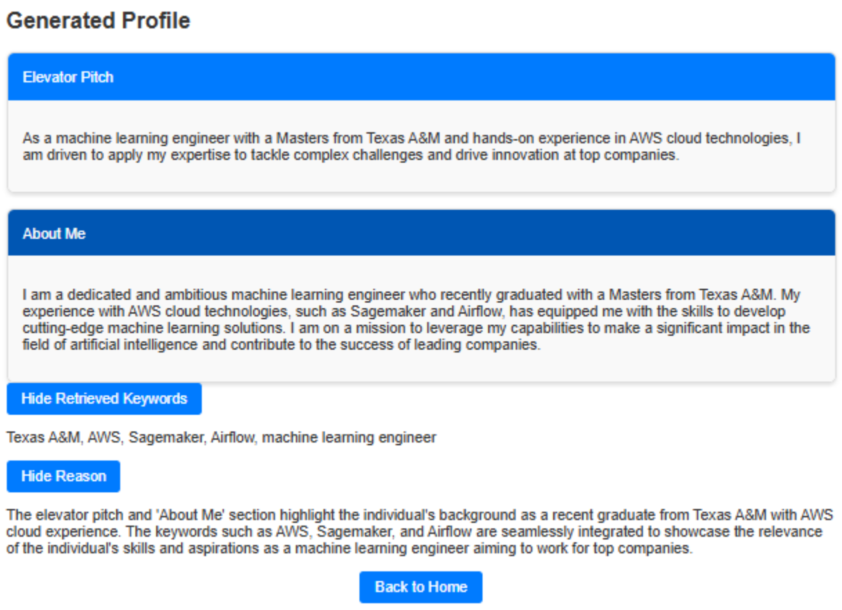
Background (optional): Graduated with Masters from Texas A&M. Have AWS cloud experience. Aim to work on top companies

Similarity Score Threshold (0-100): 9 Low threshold = Less relevent keywords

Figure 5: Sample UI Input.

5.2 Deployment

This API has been successfully integrated with the client app and the customer experience revealed overall satisfaction with the new feature. The client app is documented in this GitHub repo. A sample screenshot from the deployed app is shown in figure 7


Generated Profile

Elevator Pitch

As a machine learning engineer with a Masters from Texas A&M and hands-on experience in AWS cloud technologies, I am driven to apply my expertise to tackle complex challenges and drive innovation at top companies.

About Me

I am a dedicated and ambitious machine learning engineer who recently graduated with a Masters from Texas A&M. My experience with AWS cloud technologies, such as Sagemaker and Airflow, has equipped me with the skills to develop cutting-edge machine learning solutions. I am on a mission to leverage my capabilities to make a significant impact in the field of artificial intelligence and contribute to the success of leading companies.

Texas A&M, AWS, Sagemaker, Airflow, machine learning engineer

The elevator pitch and 'About Me' section highlight the individual's background as a recent graduate from Texas A&M with AWS cloud experience. The keywords such as AWS, Sagemaker, and Airflow are seamlessly integrated to showcase the relevance of the individual's skills and aspirations as a machine learning engineer aiming to work for top companies.

Figure 6: Sample UI Output.

6 Next Steps & Conclusion

We successfully achieved the overall goals outlined in the project proposal that planned the development of an API that generates professional and engaging profile descriptions based on given context. Furthermore, we ensured adherence to software development best practices by thoroughly testing each feature of the API. Finally, we showed example integration to the API with an experimental UI.

We acknowledge that a more realistic usage to this UI would be to deploy the app and collect the user metrics in a deployed database. This feedback can be further used to optimize the API based on the customer reviews. Also, the RAG component can be scheduled for frequent updates to ensure better UI experience. Further, more challenging tasks that combine several modalities and multiple profile elements to generate specific content or multiple contents can be of a particular interest. Also, some feature such as using this API to enhance the user resume can be easily extended. As the main scope for this project was to investigate the LLM related design choices, we didn't proceed with these plans and left them as future work.

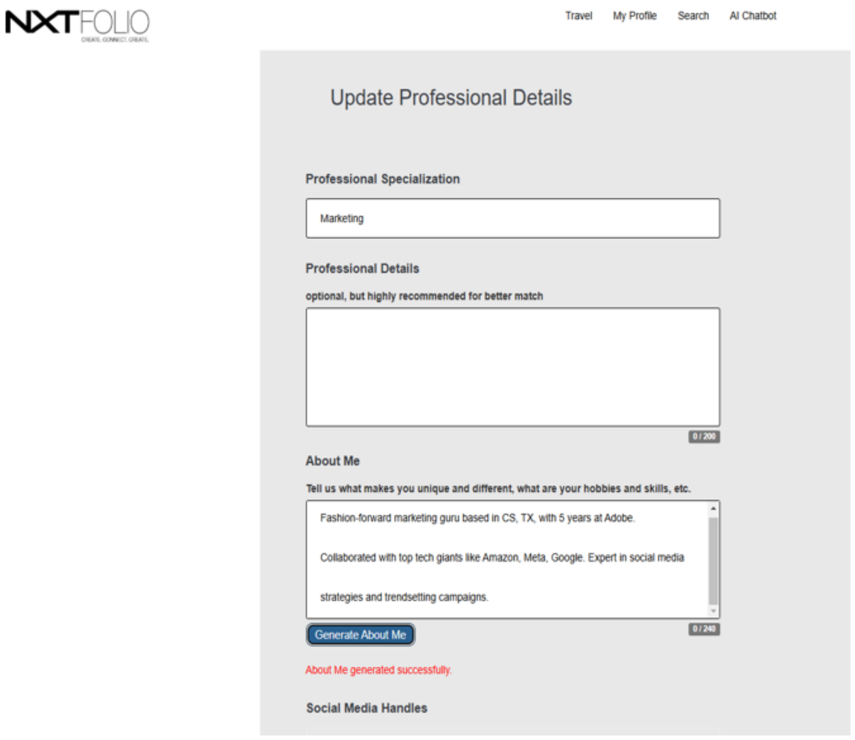
Declaration

We acknowledge that generative AI technologies, including OpenAI's ChatGPT, were utilized to assist in drafting this report. The AI tools provided support in organizing ideas, structuring sections, and refining the language of the document.

The author, Omar Khater, have thoroughly reviewed the report and confirm its content. We take full responsibility for the ideas, analyses, and conclusions presented in this report.

References

[1] AWS Workshops. Amazon bedrock workshop, 2024. Accessed: 2024-12-05.



Travel My Profile Search AI Chatbot

Update Professional Details

Professional Specialization

Marketing

Professional Details

optional, but highly recommended for better match

0 / 200

About Me

Tell us what makes you unique and different, what are your hobbies and skills, etc.

Fashion-forward marketing guru based in CS, TX, with 5 years at Adobe.

Collaborated with top tech giants like Amazon, Meta, Google. Expert in social media

strategies and trendsetting campaigns.

0 / 240

Generate About Me

About Me generated successfully.

Social Media Handles

Figure 7: Deployed App sample usage.