Resume Screening Bot

INFO 7375 Final project show and tell

Xiaoyang Chen & Swini Rodrigues

Introduction

Brief overview of the final project

Aims to present a POC of an LLM chatbot that can assist hiring managers in the resume screening process. The assistant is a cost-efficient, user-friendly, and more effective friendly, and more effective alternative to the conventional keyword-based screening methods.

Objectives and goals of the project

Powered by state-of-the-art LLMs, it can handle unstructured and complex natural language data in job descriptions/resumes while performing high-level tasks as effectively as a human recruiter.

Importance and relevance of the project to the course and industry

Despite the increasingly large volume of applicants each year, there are limited tools that can assist the screening process effectively and reliably. Existing methods often revolve around keyword-based approaches, which cannot accurately handle the complexity of natural language in human-written documents. Because of this, there is a clear opportunity to integrate LLM-based methods into this domain, which the project aims to address.

Project Description

Specific problem the project aims to solve

Many automated screening systems, such as the popular keyword matching method, have been employed to help boost the efficiency of this process. However, they often follow an over-simplistic and rigid rule-based approach by relying on a predefined set of keywords, posing clear risks of bias. Nevertheless, they also face various problems in handling the complex, context-heavy, and versatile nature of resumes written in natural language.

Detailed description of the project

The goal is to present an LLM agent system to assist hiring managers in the job-resume matching task. The key design is to integrate Retrieval Augmented Generation (RAG) to effectively retrieve the top matching resumes from a large pool of applicants and augment them to the LLM's knowledge base. Given job descriptions as queries, the LLM can use this augmented context to generate accurate and relevant assessments of applicants.

Project Description

Scope of the project

Project Goals

- Enhance answer quality for complex queries.
- Match resumes with job descriptions effectively

Functional Requirements

- Document Preprocessing
- Text Retrieval
- Text Generation
- Resume Comparison

Non-functional Requirements

- System Performance

Ensure timely retrieval and generation processes.

- Scalability

Handle large volumes of resumes and job descriptions.

- Accuracy

Provide high accuracy and relevant responses.

Technology Stack

- RAG

Combines generative agents and similarity-based retrieval.

- LLM

For sub-query and response generation

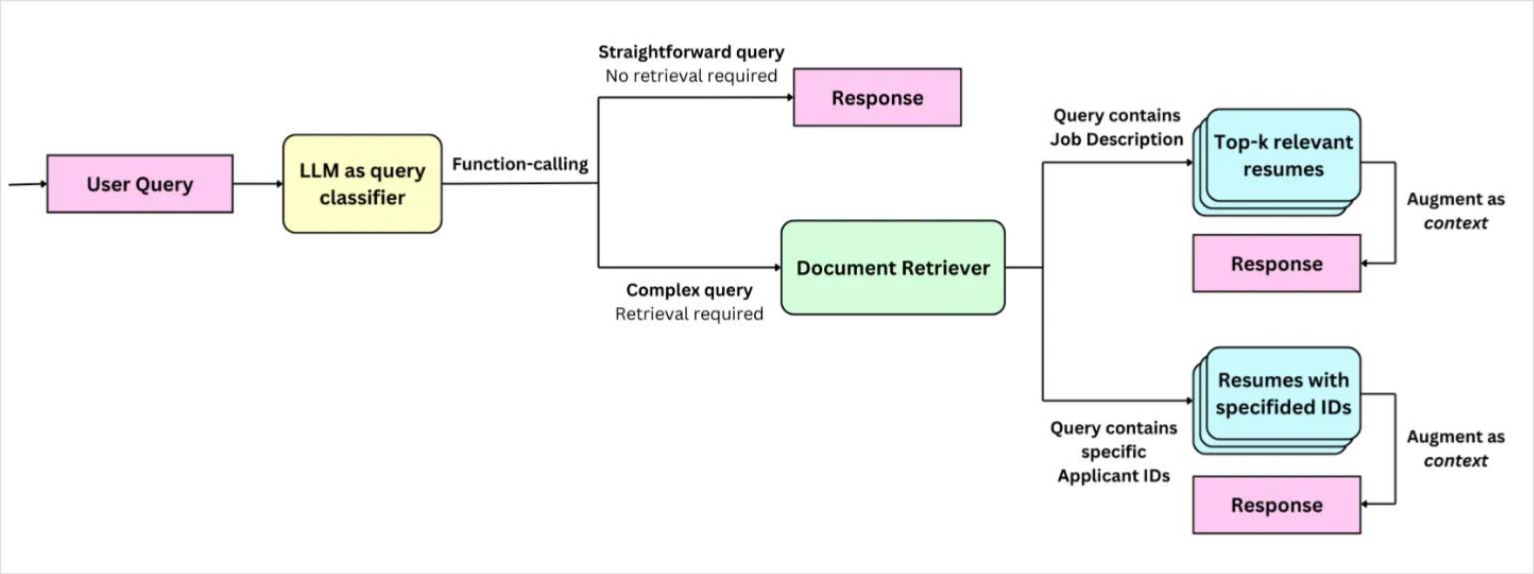
- Vector Storage

For similarity retrieval

Boundaries

- Process only text-based resumes and job descriptions.
- Exclude handling of multimodal data (e.g., images, videos).

Project Architecture



Technologies and tools used

- langchain, openai, huggingface: RAG pipeline and chatbot construction.

- faiss: Vector indexing and similarity retrieval.

- streamlit: User interface development.

Data Collection and Preprocessing

Source and nature of the data

Job Title and Job Description Dataset from kaggle

Data preprocessing techniques used

only redundant spacing, line breaks, breaks, and invalid non-ASCII characters for better readability readability during the assessment assessment phase

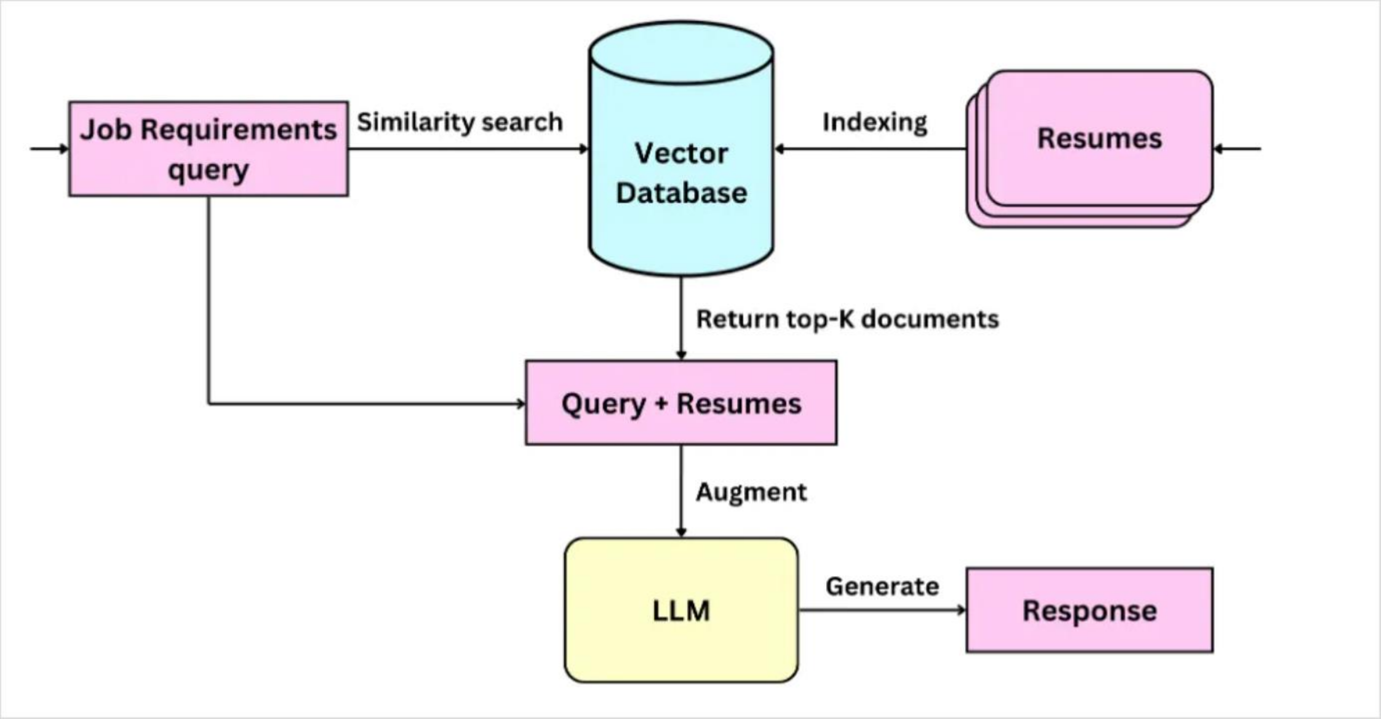
Ground truth and noise generate

Use gpt-3.5 to generate two distinct types of resumes for each job description. The first one is a near-perfect match to the job description and serves as the ground truth answer, while the other is less relevant and is included primarily as noise to the test data.

Final dataset

A list of 500 job descriptions as queries with the corresponding highly suitable resumes as ground truth. This test set can be utilized to evaluate the proposed model in finding candidates similar to the ground truth resumes corresponding to each of the 500 job descriptions in a large pool of 1000 applicants.

RAG Pipeline Implementation



Performance Metrics

Objective

- Evaluate the model in the job-resume matching task.
- Task

  Find the most suitable resume for each job description among a heterogeneous database of 1000 resumes.

Methods

- Use synthetic data. Generate two types of resumes from real-world job descriptions
- Ground truth resume

  Highly suitable applicant’s profile to job descriptions.
- Noise resume

  Less relevant resume to job descriptions.

Metrics

- Semantic Similarity

  Semantic similarity is determined by the cosine similarity score between the vector representations of the selected resume and the ground truth resume.
- Accuracy

  The accuracy measures the portion of correct selections of ground truth resumes, which can be useful in determining whether the system can locate the exact matching resume.

$$\text{Similarity} = \text{Cos Similarity}(\text{Embedding}_{\text{Selected Resume}}, \text{Embedding}_{\text{Ground Truth}})$$

$$\text{Accuracy} = \frac{\text{Correctly Predictions}}{\text{All Predictions}} = \frac{\sum_{i=0}^{n} 1(\text{Selected}_i = \text{Ground Truth}_i)}{n}$$

Methods to Improve Metrics

Change embedding model

utilize text embedding models that are specialized in recruitment-related contexts

more advanced chunking strategies

utilize more advanced chunking strategies such as content-based chunking. For instance, resumes can be separated into chunks by sections (experience level, skills, etc.) to eliminate the overlapping noises between each section.

larger datasets for training

creating larger and more authoritative recruitment datasets for training cause the complex job description in the real world

Future work

use Rag-fusion framework

The key idea of RAG fusion is to first deconstruct the original input queries into smaller but more focused sub-queries.
For each query, the retriever uses similarity-based retrieval to get a list of relevant documents. A specialized algorithm then combines and re-ranks these lists. The final result is a new ranked list of the most relevant documents for the original complex query.

combining RAG with fine-tuning

Future research may experiment with combining RAG with fine-tuning to assist the LLM model in learning more specific contexts and answer paradigms in real-life scenarios.

Conclusion

The proposed system is a RAG Fusion pipeline integrated with a GPT-3.5-turbo agent aiming to assist recruiters in matching job descriptions with suitable resumes. This experiment demonstrates the potential of the proposed RAG systems in resume screening and highlights the need for further research.

Any questions?

Github repo: https://github.com/SlipRiders/Resume-Screening-Bot.git