

INFO 7375 FINAL Resume Screening RAG Bot

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 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

- 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.

- 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.

- Project Scope

Project Goals:

- Enhance answer quality, especially for complex and ambiguous queries.
- Effectively match resumes with job descriptions in recruitment scenarios.

Functional Requirements:

- **Document Preprocessing:** Split documents into smaller text chunks, generate sub-queries, and embed resume chunks and sub-queries into representation vectors.
- **Text Retrieval:** Utilize vector storage for similarity retrieval, return the most relevant text chunks, and re-rank them into a new document list.

- **Text Generation:** Use LLM to combine retrieved text chunks and original query, generating accurate and contextually relevant responses.
- **Resume Comparison:** Compare multiple resumes with specified IDs and provide recommendations.

Non-functional Requirements:

- **System Performance:** Ensure retrieval and generation processes complete within reasonable time frames.
- **Scalability:** Handle large volumes of resumes and job descriptions.
- **Accuracy:** Generate responses with high accuracy and contextual relevance.

Technology Stack:

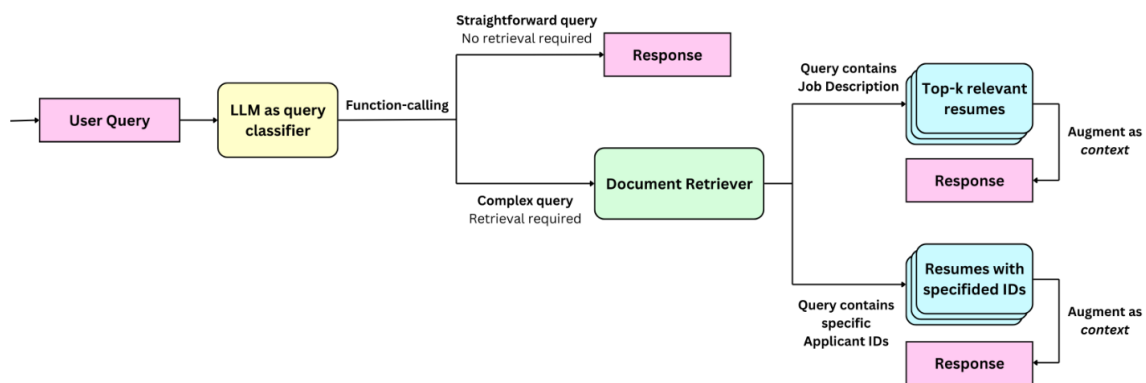
- **RAG:** Combining generative agents and similarity-based retrieval.
- **LLM:** For sub-query generation and response generation (e.g., OpenAI's models).
- **Vector Storage and Embedding Models:** For similarity retrieval (e.g., faiss).

Project Boundaries:

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

Project Architecture:

- Diagram of the 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 [Kaggle](#)
- Data preprocessing techniques used

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

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.

The final test set to evaluate the system is the 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.

System Prompt:

You are an AI assistant that helps create resumes for a given job description.

Generate 2 resumes for each job description so that one resume is an almost perfect match, while the other resume is only slightly relevant.

Use a combination of skills, different industry/project work experience, education, and certifications to produce resume data.

You may add some KPIs to make work experience realistic.

Do not include any note or explanation of how you generate the resumes.

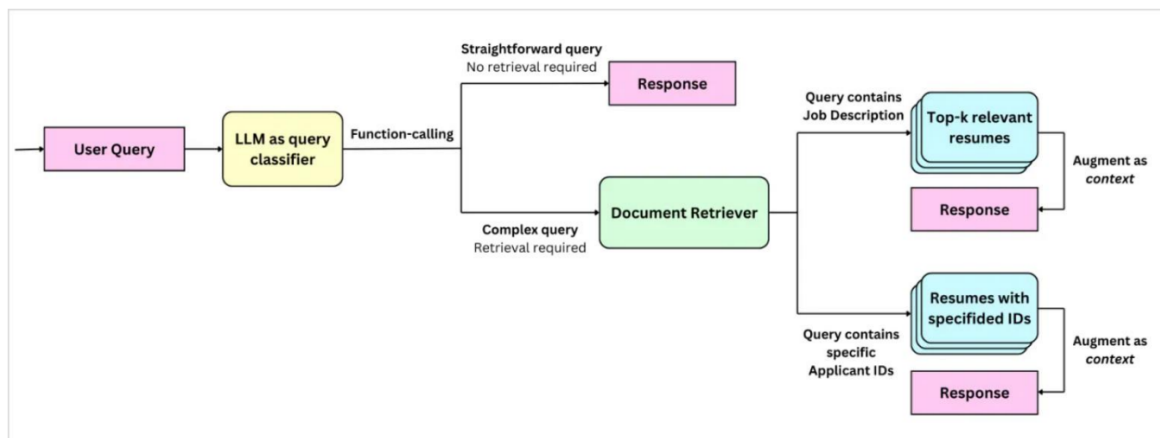
User Prompt:

Here is the Job Description (Note that all required skills may not be present in the resume and some nonrelevant details can be present).

The length of a resume should only be between 200 and 500 words.

{job_description}

Project Architecture



Fine-tuning

Data Acquisition and Preparation

1. Data Collection:

- The datasets used for fine-tuning included job descriptions, resumes, and a test set. These were obtained from structured CSV files stored in Google Drive.

2. Data Loading:

- The datasets were read into pandas DataFrames to facilitate easy manipulation and merging.

3. Data Merging:

- Job descriptions and resumes were merged to form a comprehensive training dataset. Each job description was paired with each resume to create a diverse set of training examples.

4. Data Sampling:

- To manage computational resources efficiently, a random sample of 2000 data points was taken from the merged dataset for the fine-tuning process.

5. Label Assignment:

- Labels were assigned to the dataset. In this case, a default label of 0.0 was used, indicating a binary or regression task.

Model and Tokenizer Selection

1. Pre-trained Model:

- A pre-trained model, specifically `sentence-transformers/all-MiniLM-L6-v2`, was chosen for its balance of performance and efficiency.

2. Tokenizer Initialization:

- The corresponding tokenizer for the pre-trained model was initialized to handle text preprocessing.

Data Preprocessing

1. Tokenization:

- A preprocessing function was written to tokenize the input texts (job descriptions and resumes). This function also ensured that the labels were converted to the appropriate format (floating-point).

2. Dataset Conversion:

- The tokenized data was converted into a format compatible with the transformer model's training process.

Training Configuration

1. Training Arguments:

- The training configuration included setting parameters such as the learning rate, batch size, number of epochs, and weight decay. These parameters were chosen to optimize the fine-tuning process while preventing overfitting.

2. Hardware Utilization:

- The training process was configured to utilize available hardware efficiently, opting for GPU acceleration if available, otherwise defaulting to CPU.

Model Fine-tuning

1. Trainer Initialization:

- A `Trainer` class from the `transformers` library was instantiated. This class simplifies the training loop and integrates seamlessly with the datasets and model.

2. Training Execution:

- The fine-tuning process was executed, involving multiple epochs of training where the model learned to map job descriptions to resumes.

Model Evaluation and Saving

1. Model Evaluation:

- During training, the model's performance was monitored to ensure that it was learning effectively from the data.

2. Model Saving:

- Upon completion of the training process, the fine-tuned model and tokenizer were saved for future use. The models were also backed up to Google Drive to ensure

accessibility.

Functionality and test cases

Currently, this demo app supports the following two types of queries 💬:

1. Enter the job description to find the most suitable resume

Example:

Find a data Scientist with python experience



Find a data Scientist with python experience



Based on the provided resumes and the job description of finding a Data Scientist with Python experience, the best candidate would be **Applicant ID 979 - Samantha Lee**.

Here's why Samantha Lee is the best fit for the Data Scientist position with Python experience:

1. **Relevant Experience:** Samantha has 3 years of experience as a Data Scientist, which involves developing advanced algorithms, executing statistical techniques, and working with Python libraries like Pandas and Numpy for data processing. This experience aligns well with the requirements for the Data Scientist role.
2. **Python Proficiency:** Samantha's experience with Python libraries such as Pandas, Numpy, and NLTK showcases her strong Python skills, which are essential for a Data Scientist role that requires data processing, analysis, and modeling.
3. **Data Analysis Skills:** Samantha's experience in analyzing data, creating visualizations, and working with analytical models for large datasets demonstrates her ability to handle complex data sets effectively, a crucial skill for a Data Scientist.
4. **Educational Background:** Samantha holds a Master of Science in Data Science, which further solidifies her qualifications for the Data Scientist position.
5. **Certifications:** Samantha is an AWS Certified Cloud Practitioner, indicating her proficiency in cloud technologies, which could be beneficial for a Data Scientist role that involves working with cloud platforms like AWS.

Overall, Samantha Lee's combination of relevant experience, Python proficiency, data analysis skills, educational background, and certifications make her the most suitable candidate for the Data Scientist position with Python experience.

and NLTK showcases her strong Python skills, which that requires data processing, analysis, and model

3. **Data Analysis Skills:** Samantha's experience in ana working with analytical models for large datasets d data sets effectively, a crucial skill for a Data Scient

4. **Educational Background:** Samantha holds a Maste solidifies her qualifications for the Data Scientist p

5. **Certifications:** Samantha is an AWS Certified Cloud cloud technologies, which could be beneficial for a with cloud platforms like AWS.

Overall, Samantha Lee's combination of relevant exper skills, educational background, and certifications make Data Scientist position with Python experience.

Verbosity

Total time elapsed: 2.151 seconds

A job description is detected. The system defaults to

Using Generic RAG to retrieve...

Returning top 5 most similar resumes...

Resume 1 ▾ Resume 2 ▾ Resume 3 ▾

Extracted query: `{'job_description': 'Data Scientist with Python experience'}`

Generated questions: `['Data Scientist with Python experience']`

Document re-ranking scores: `{'701': 0.02, '325': 0.0196078431372549, '979': 0.019230769230769232, '885': 0.018867924528301886}`

Applicant ID 805 ():

Education:

- MS in Computer Science, with coursework and projects in machine learning and data analysis
- Bachelor's degree in Computer Engineering

Experience:

- 4+ years of experience in machine learning, with a focus on feature engineering, training, validation, scaling, deployment, scoring, monitoring, and feedback loop
- Analyzed complex problems and translated them into analytical approaches using supervised and unsupervised machine learning techniques, including classification, forecasting, anomaly detection, and text mining, using decision trees, time series analysis, bagging and boosting algorithms, neural networks, and deep learning
- Experienced with analytical programming languages, tools, and libraries, specifically the Python ecosystem, and SQL and relational databases
- Good understanding of programming best practices and building for re-use
- Collaborated in a team environment to support growth and development

Certifications:

- Certified Machine Learning Engineer
- Certified Data Scientist

KPIs:

- Successfully developed and deployed a machine learning model that improved customer acquisition by 25%

2. Enter a specific ID to find a specific resume

Example:

compare applicant ID 1 with applicant ID 23



compare applicant ID 1 with applicant ID 23



Applicant ID 1 is an experienced iOS Developer named Jane Smith based in Pune, India. She has a Bachelor of Engineering in Computer Science and has worked as an iOS Developer at two different companies, LMN Company and PQR Company. Jane has a strong command over Swift and objective C, good knowledge of interface builder, Core animation, and Object oriented concepts. She is also familiar with frameworks & web technologies and has excellent communication skills.

Applicant ID 23 is a Backend Developer with experience in developing websites and applications using HTML, CSS, JavaScript, jQuery, and PHP. They are proficient in building WordPress CMS websites, customizing themes and plugins, and familiar with MySQL database and Ooops Concepts. This applicant has worked as a Backend Developer at DEF Company.

In comparing the two applicants, Jane Smith (Applicant ID 1) is specialized in iOS development with a focus on Swift and objective C, while Applicant ID 23 is a Backend Developer with expertise in web development technologies like HTML, CSS, JavaScript, and PHP, with a focus on WordPress CMS. The decision on which applicant to proceed with would depend on the specific needs and requirements of the role you are looking to fill.

Verbosity

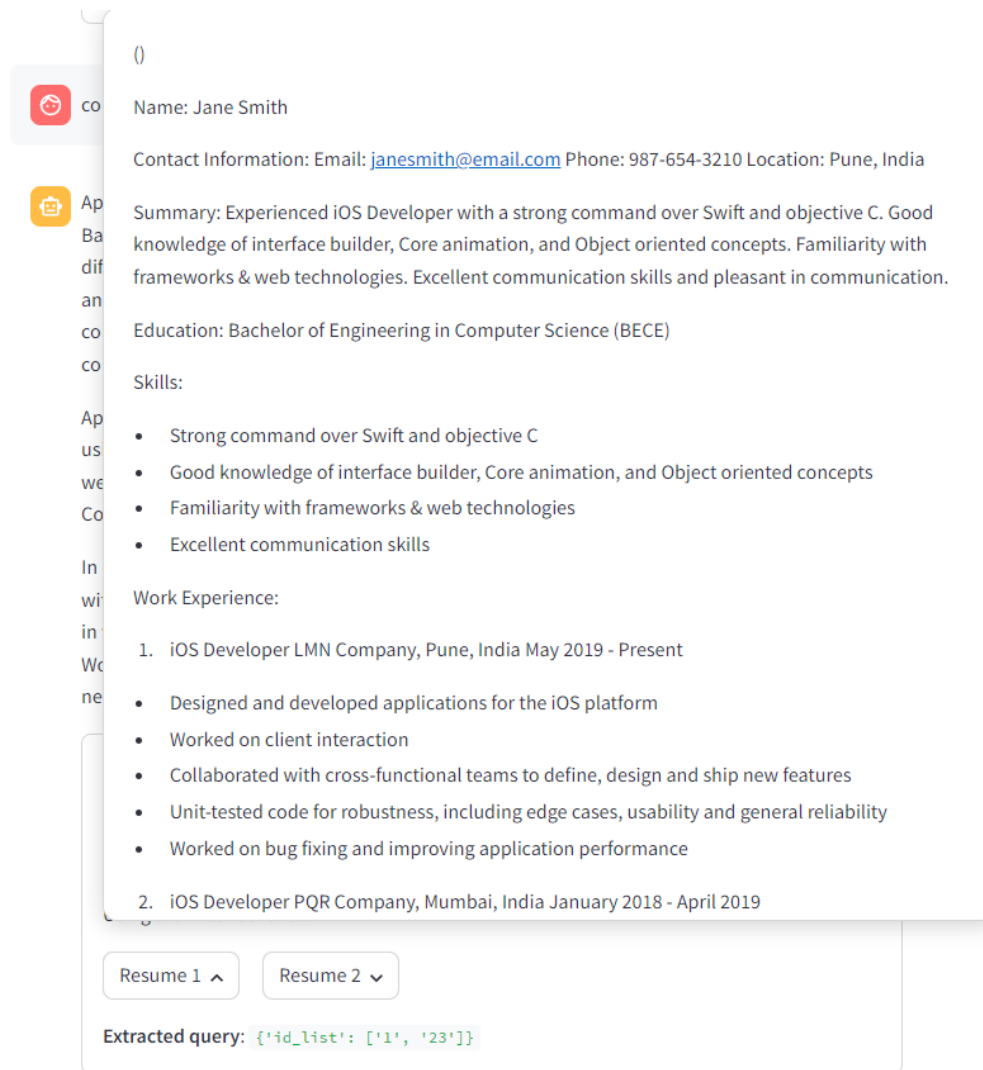
Total time elapsed: 0.89 seconds

Applicant IDs are provided. The system defaults to using exact ID retrieval...

Using the ID to retrieve...

Resume 1 ▾ Resume 2 ▾

Extracted query: `{'id_list': ['1', '23']}`



Performance Metrics

To evaluate RAG systems effectively, we use a framework called Retrieval Augmented Generation Assessment (RAGAs). This framework aims to measure the two dimensions of a RAG system, the retriever and the generator. To achieve this, RAGAs utilizes an LLM agent to analyze the context, the question, the RAG system's response, and the ground truth labels to extract the components necessary for computing each metric. In the context of this experiment, with the resources available, the gpt-3.5-turbo-1106 model is employed as the LLM for RAGAs evaluation.

Additionally, the following metrics are chosen: Context Recall (CR), Context Precision (CP), Faithfulness (FA), and Answer Similarity (AS), Semantic Similarity(SS), Accuracy. The selected metrics all return a score ranging from 0 to 1, with higher values denoting better performance. In the experiment, the metrics are calculated for each of the 500 entries in the test set. The mean values of these scores are then computed to assess the average performance of the model over the entire test set.

Context Recall (CR) CR measures the extent to which the retrieved context aligns well with the ground truth answer. In the context of the experiment, this essentially computes the degree of relevance of the retrieved resumes to the ground truth resume. This metric can directly measure the retriever component's ability to search for candidates similar to the ideal resume.

Context Precision (CP) CP measures whether the ground-truth relevant details present in the contexts are ranked higher or not. In an ideal scenario, the relevant documents should be at the top ranks. CP can evaluate whether the retriever component ranks resumes with more relevant details to the ground truth higher among all documents. This helps with the assessment of the precision of the retriever component's ranking ability.

Faithfulness (FA) FA computes the consistency of the answer to the provided context. According to RAGAS' definition, the generated answer is considered faithful if claims made in the answer can be inferred from the context. FA measures the generator's faithfulness and consistency to the retrieved context, which helps to determine whether the LLM component is prone to hallucinations or not. This is especially important in a knowledge-intensive task like resume screening, where the processing of candidates' information is often required to be precise.

Answer Similarity (AS) AS measures the semantic similarity between the generated answer and the ground truth. This can be utilized to assess the LLM's summarization quality of the selected resumes.

Semantic Similarity Semantic similarity is determined by the cosine similarity score between the vector representations of the selected resume and the ground truth resume. In essence, the metric measures the relevance of the selected resume to the ground truth resume. This helps to assess the ability to retrieve resumes relevant to the ground truth of the RAG system.

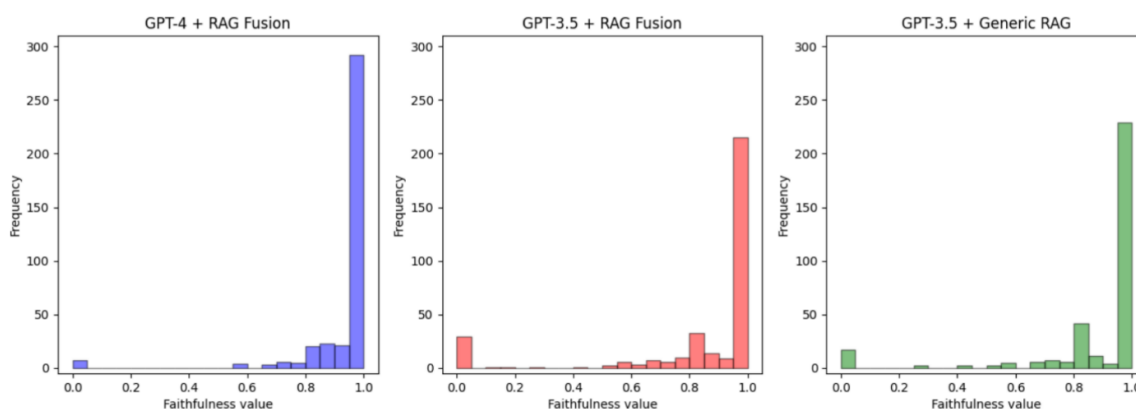
Accuracy The accuracy measures the portion of correct selections of ground truth resumes, which can be useful in determining whether the system can locate the exact matching resume.

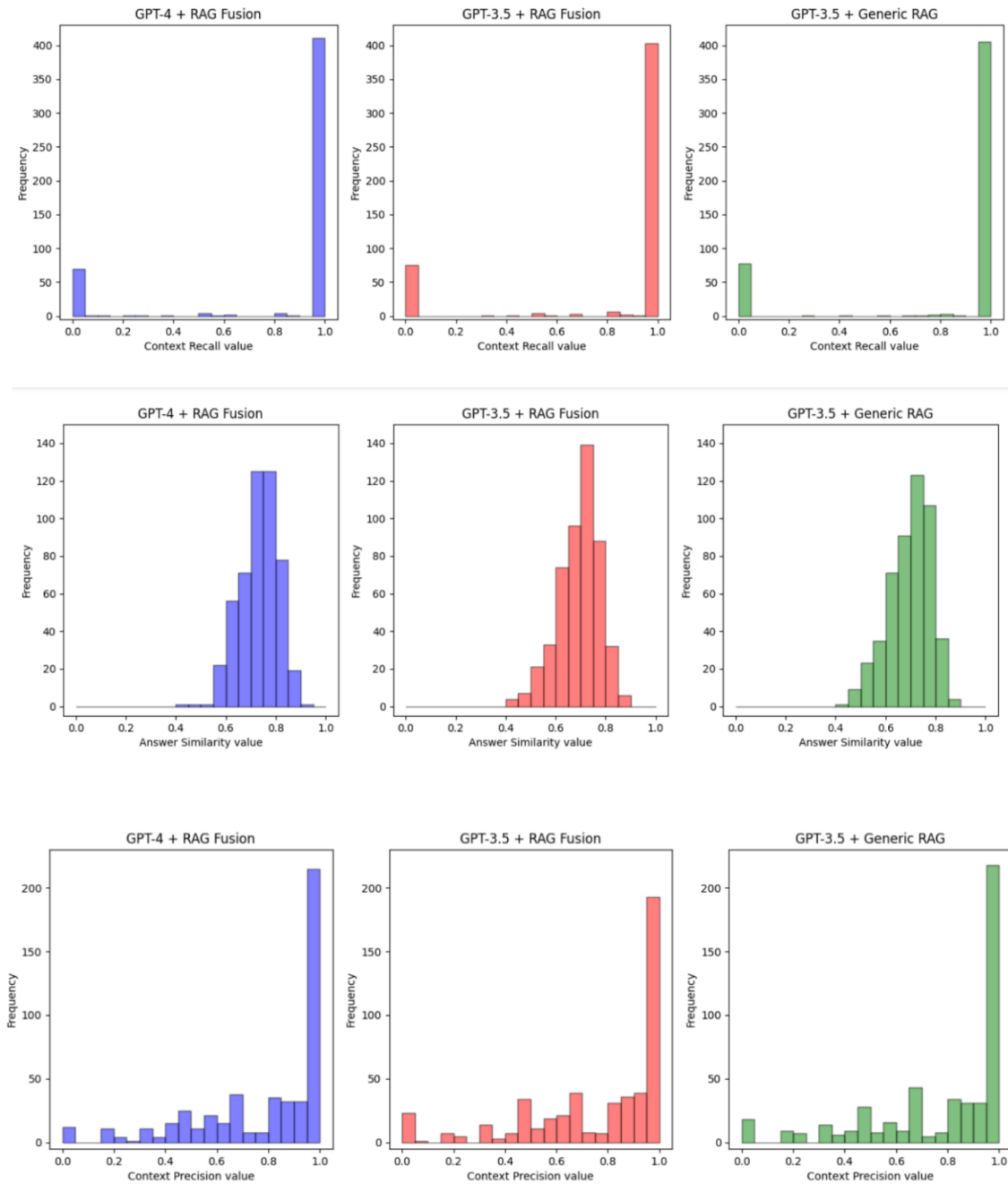
The first baseline is a RAG Fusion system utilizing gpt-3.5-turbo-1106 as the generator. It employs the same LLM agent for sub-questions generation tasks. Essentially, the baseline simply utilizes a less powerful model than GPT-4 as the LLM agent for multiple tasks, which can provide insights into how the LLM agent choice affects resume selection quality.

The second baseline system is a Naive RAG framework utilizing gpt-3.5-turbo-1106 as the generator. Besides the clear difference in the LLM agent choice, this RAG system utilizes just the generic similarity-based retrieval approach with no RRF re-ranking. It can be a meaningful baseline to understand whether a less powerful LLM agent with a simpler retriever approach can perform comparably with the more advanced models.

Metrics calculate results

Calculate the scores of the 4 RAGAS metrics and two other metrics for each model/framework and plot them in graphs. Extract the predicted resume ID from test_result_df and compare it with the target resume ID to calculate the accuracy.





Models	CR	CP	FA	AS
GPT-4 + RAG Fusion	0.843	0.793	0.945	0.733
GPT-3.5 + RAG Fusion	0.837	0.769	0.846	0.694
GPT-3.5 + Generic RAG	0.837	0.787	0.887	0.696

Overall, the distribution of the scores for the models for every metric is relatively similar. The similarity suggests a consistent performance of each model across all metrics, indicating that the evaluation process is relatively stable and balanced.

In addition, it is also clear to observe the same distribution trend, which is skewed towards higher values. For FA, CR, and CP particularly, a significantly large number of values is observed to be a perfect score of 1. In general, this result suggests that the performance of all models is mostly satisfactory so far.

For a better look at the performance of each component of each model over the whole test set, Table above presents the mean CR, CP, FA, and AS scores for the proposed model and the baseline models over the 500 job descriptions.

The CR of the proposed model to the baselines is slightly higher than the baselines. A higher CR means that the retriever is better at identifying and retrieving documents that are relevant to the ground truth resume. Even though this suggests that the GPT-4 + RAG Fusion retriever is more effective, it is clear that this difference is marginal. Despite a more advanced re-ranking approach, the proposed model does not provide a significant improvement to the recall of the context.

In terms of CP score, the GPT-4 + RAG Fusion model also performs relatively similarly to the Generic RAG model. Once again, this may reveal that the Fusion retriever model may not necessarily improve the final ranking results. Interestingly, the GPT-3.5 + RAG Fusion model exhibits a 2% decrease in performance compared to the other models. This can be due to various factors such as the GPT-3.5's less advanced language comprehension, which leads to suboptimal sub-queries generation.

On the other hand, the FA score of the proposed model significantly outperforms the baselines. This is expected, given that the GPT-4 model is substantially more powerful than GPT-3.5 in reading comprehension. Therefore, the generated answer can be considerably more consistent and faithful to the provided context. In addition, this result implies that GPT-4 tends to hallucinate far less than the LLM generators of the baseline models.

Likewise, the AS score of the proposed model is notably higher than the baselines. This outcome may be a result of a combination of the proposed model's more effective summarization of details and better identification of similar resumes. Once again, this may be because of the advanced comprehension and contextual understanding capabilities of the GPT-4 model, allowing it to generate more concise analyses and summaries of resumes.

To summary, even with a more advanced design, the retriever component of RAG Fusion models did not yield notable improvements in ranking and retrieving resumes. Moreover, in some cases GPT-3.5 is utilized for RAG Fusion's sub-queries generation, the outcome of document ranking is even slightly less effective than Naive RAG's generic retriever. On the other hand, the solid improvements in FA and AS scores of the proposed model can mostly be attributed to the GPT-4 LLM generator. Being a considerably more powerful LLM than GPT-3.5, this model can provide summaries and explanations of selected resumes that closely follow the details in the provided context.

To better assess the resume selection outcome for the system as a whole, it is necessary to investigate the rest of the metrics. Table below displays the average similarity and accuracy, which are obtained by comparing the selected resume to the ground truth resume.

Models	Accuracy	Similarity
GPT-4 + RAG Fusion	0.558	0.885
GPT-3.5 + RAG Fusion	0.426	0.851
GPT-3.5 + Generic RAG	0.436	0.852

Given that multiple candidates might match the ideal profile at the same time, the accuracy scores of the models can be considered satisfactory.

Among 500 job descriptions with up to 1000 resumes, the models can identify around 120 to 280 cases. Once again, it is visible that the selection accuracies of the two baseline models are relatively similar. Meanwhile, the proposed model stands out with a significant improvement in accuracy, suggesting its higher effectiveness in accounting for more nuanced factors in the job requirements to identify the exact matching resume.

On the other hand, the selected resumes for all models exhibit a high degree of similarity to the ground truth. On average, the selected resumes by each model are 85% to 89% similar to the ground truth. This indicates that the RAG systems can effectively determine highly similar candidates among a large database of resumes. It is noteworthy that GPT-4 achieves the highest resume similarity score of up to around 89%. This suggests that the resumes selected by the proposed model are generally more suitable.