

I tried to seek resume screening examples that already exist.

I found

<https://github.com/Hungreeee/Resume-Screening-RAG-Pipeline>

I tried it on and see how it went.

1. Git clone and download requirements \*I already installed on different day but forgot to take a screenshot, so I just cd Resume-Screening-RAG-Pipeline today.

```
celiaho@Cs-Mac ~ % cd Resume-Screening-RAG-Pipeline
celiaho@Cs-Mac Resume-Screening-RAG-Pipeline % pip install -r requirements.txt
```

And then I streamlit run demo/interface.py and it directed me to a website.

```
celiaho@Cs-Mac Resume-Screening-RAG-Pipeline % streamlit run demo/interface.py

Welcome to Streamlit!

If you'd like to receive helpful onboarding emails, news, offers, promotions,
and the occasional swap, please enter your email address below. Otherwise,
leave this field blank.

Email:

You can find our privacy policy at https://streamlit.io/privacy-policy

Summary:
- This open source library collects usage statistics.
- We cannot see and do not store information contained inside Streamlit apps,
  such as text, charts, images, etc.
- Telemetry data is stored in servers in the United States.
- If you'd like to opt out, add the following to ~/.streamlit/config.toml,
  creating that file if necessary:

[browser]
gatherUsageStats = false

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.2:8501

/opt/homebrew/anaconda3/lib/python3.11/site-packages/pandas/core/array/masked.py:60: UserWarning: Pandas requires version '1.3.6' or newer of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
.gitattributes: 100% | 1.23k/1.23k [00:00<00:00, 995kB/s]
1_Pooling/config.json: 100% | 190/190 [00:00<00:00, 522kB/s]
README.md: 100% | 10.7k/10.7k [00:00<00:00, 13.9MB/s]
config.json: 100% | 612/612 [00:00<00:00, 1.49MB/s]
config_sentence_transformers.json: 100% | 116/116 [00:00<00:00, 397kB/s]
data_config.json: 100% | 39.3k/39.3k [00:00<00:00, 39.8MB/s]
model.safetensors: 100% | 96.9M/96.9M [00:07<00:00, 11.7MB/s]
pytorch_model.bin: 100% | 96.9M/96.9M [00:06<00:00, 11.2MB/s]
sentence_bert_config.json: 100% | 53.0/53.0 [00:00<00:00, 185kB/s]
special_tokens_map.json: 100% | 112/112 [00:00<00:00, 366kB/s]
tokenizer.json: 100% | 466k/466k [00:01<00:00, 284kB/s]
tokenizer_config.json: 100% | 380/380 [00:00<00:00, 1.22MB/s]
train_script.py: 100% | 13.2k/13.2k [00:00<00:00, 23.3MB/s]
vocab.txt: 100% | 232k/232k [00:00<00:00, 445kB/s]
modules.json: 100% | 349/349 [00:00<00:00, 1.22MB/s]
2024-05-15 15:37:49.139 Uncaught app exception
Traceback (most recent call last):
  File "/opt/homebrew/anaconda3/lib/python3.11/site-packages/streamlit/runtime/scriptrunner/script_runner.py", line 600, in _run_script
    exec(code, module.__dict__)
  File "/Users/celiaho/Resume-Screening-RAG-Pipeline/demo/interface.py", line 207, in <module>
    query_type = llm.query_classification(user_query)
  File "/Users/celiaho/Resume-Screening-RAG-Pipeline/demo/llm_agent.py", line 176, in query_classification
    response = self.llm.invoke([system_message, user_message])
```

In there, I can see the introduction and how to get it started. I entered my OpenAI API Key

Control Panel

OpenAI's API Key

Generic RAG

GPT Model

gpt-3.5-turbo-1106

Upload resumes

Drag and drop file here

Limit 200MB per file - CSV

Browse files

celia-evelyn-horyono-cv...  
0.0B

Clear conversation

Resume Screening GPT

Introduction

The system is a RAG pipeline designed to assist hiring managers in searching for the most suitable candidates out of hundreds of resumes more effectively.

The idea is to use a similarity retriever to identify the most suitable applicants with job descriptions. This data is then augmented into an LLM generator for downstream tasks such as analysis, summarization, and decision-making.

Getting started

1. To set up, please add your OpenAI's API key.

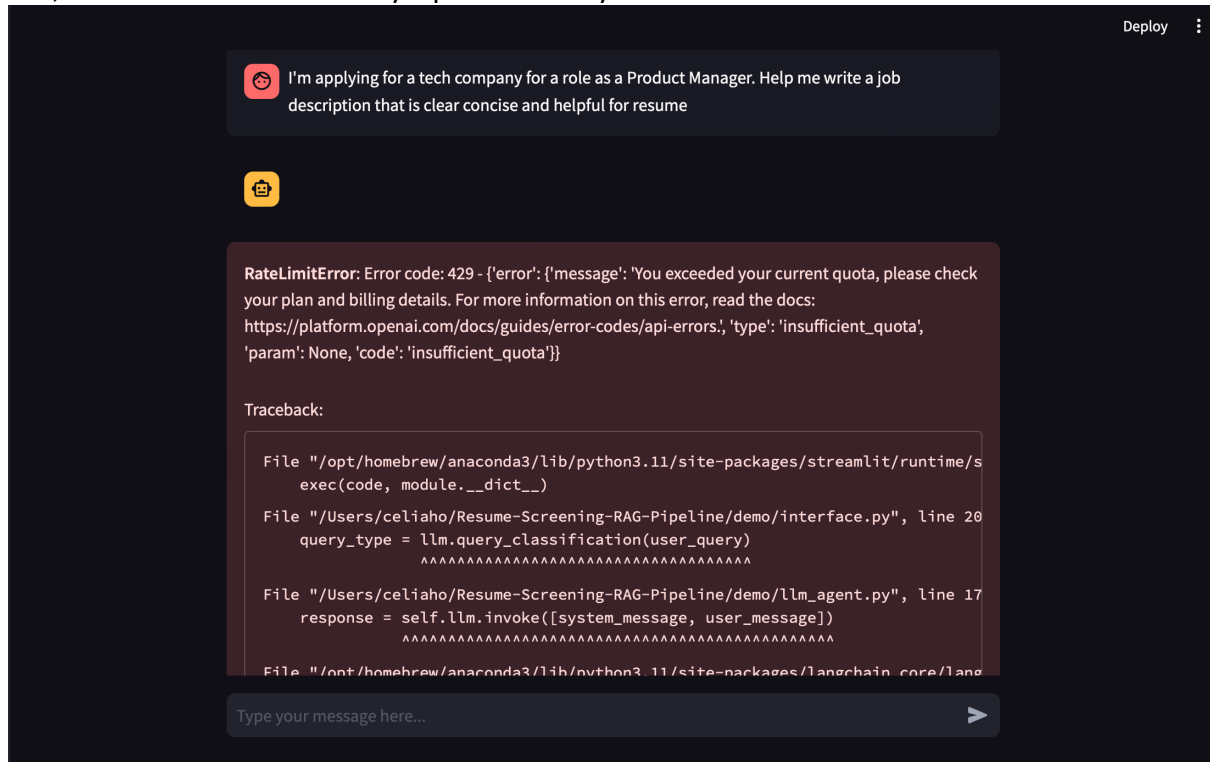
2. Type in a job description query.

Hint: The knowledge base of the LLM has been loaded with a pre-existing vectorstore of resumes to be used right away. In addition, you may also find example job descriptions to test [here](#).

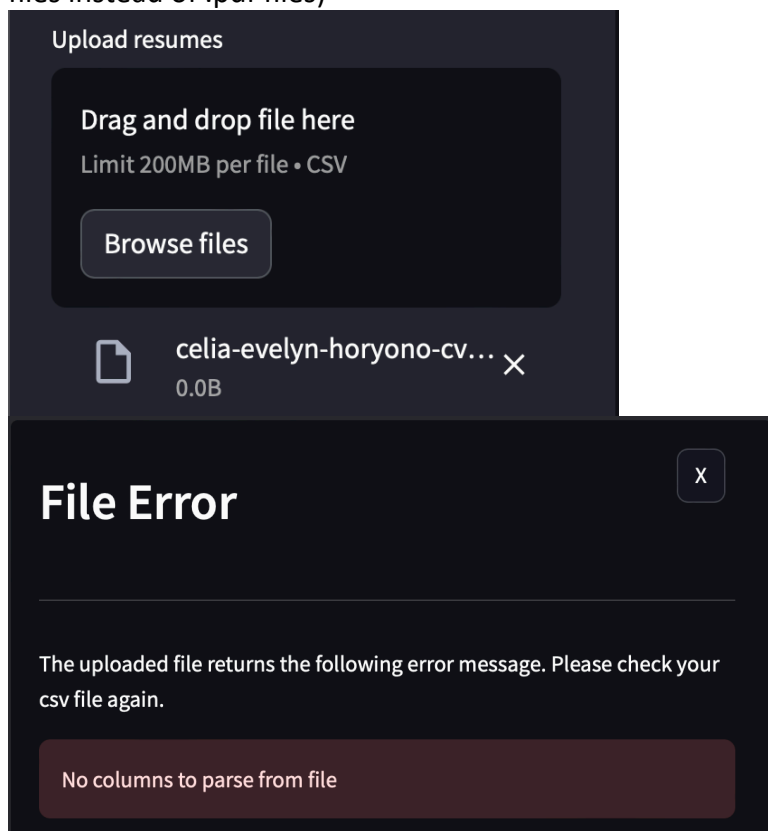
Please make sure to check the sidebar for more useful information.

Type your message here...

But, it didn't work because my OpenAI API key is the free one.



I also uploaded by CV for screening but it showed another error. (The CV needs to be in .csv files instead of .pdf files)



I then seek for different way to train my LLM. I used distilgpt2 instead of llama2

```
[7]: model_name = "distilgpt2"

[8]: from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained(model_name)

[9]: def check_skills(resume_text, required_skills):
    matched_skills = [skill for skill in required_skills if skill in resume_text]
    return matched_skills

    def check_experience(resume_text, relevant_experience):
        return experience_matches

    def check_education(resume_text, required_education):
        return education_matches

    def check_language(resume_text):
        return language_issues

[10]: def analyze_resume_with_distilgpt2(resume_text):
    inputs = tokenizer(resume_text, return_tensors="pt")
    outputs = model.generate(**inputs)
    analysis_result = tokenizer.decode(outputs[0], skip_special_tokens=True)
    return analysis_result
```

The reason why I use distilgpt2 instead of llama2 is because llama2 required access.

Your request to access this repo has been successfully submitted, and is pending a review from the repo's authors.

Another reason is because if I want to use llama2, I have to log in from huggingface but I always encounter an error.

I'm still hoping I can still use llama2, but for now, distilgpt2 is my way out. I'm currently using Week7 homework as reference.