Project 9: A Multi-Agent Generative Resume and Cover Letter Assistant with Skill Matching

Project Description

The "Multi-Agent Generative Resume and Cover Letter Assistant with Skill Matching" is an AI tool designed to help users create tailored resumes and cover letters based on job descriptions and personal skills. The tool leverages multi-agent collaboration within CrewAI to automate different aspects of the application process, including generating tailored content, performing skill matching, and providing feedback on document quality.

1. **Content Generation**: A generative AI agent (e.g., GPT-4) creates personalized resume and cover letter content, aligning with the job description and user profile.
2. **Skill Matching**: An NLP-based agent identifies and highlights relevant skills by analyzing the job description and comparing it with the user's skills.
3. **Feedback and Refinement**: A feedback agent reviews the generated content, offering suggestions for improvement in structure, clarity, and relevance.
4. **User Interface**: An interface allows users to input job descriptions, update their skills, view generated documents, and receive feedback.

Important Tutorials:

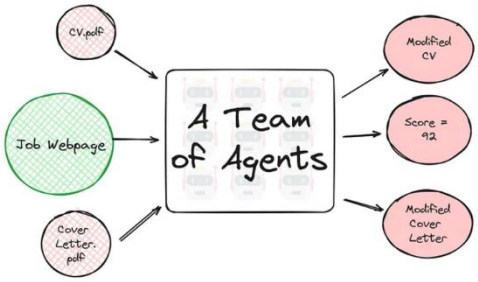
- https://towardsdatascience.com/a-comprehensive-guide-to-collaborative-ai-agents-in-practice-1f4048947d9c
- https://www.deeplearning.ai/short-courses/multi-ai-agent-systems-with-crewai/

Note: You have the flexibility to select the models and datasets that align best with your project's goals. The steps and tools outlined here are merely suggestions to guide you; feel free to tailor, modify, or expand upon these ideas to make the project uniquely your own.

IMPORTANCE : 30% Final project (Project Proposal and Implementation 10%, Presentation 10%)!!!!!

DEADLINE: 18th and 19th of December!!!!

Draft – Plan



1. Build functionality for uploading **input files** such as job descriptions, CVs, and cover letter descriptions. As one of the system requirements is to implement **User interface** as a solution design for it could be an HTML page (use FLASK to connect with backend) **or** we can do it using terminal through the datafile folder.

    a. Develop a drag-and-drop or upload feature for users to upload:
        i. Job descriptions in PDF, DOCX, or plain text.
        ii. CVs in DOCX or PDF formats.
        iii. Additional descriptions or cover letter instructions.
    b. Implement a file preprocessing system to convert uploaded documents into plain text for easier analysis.
    c. Extract relevant sections from each input:
        i. For job descriptions: extract required skills, job roles, and key responsibilities.
        ii. For CVs: extract skills, work experience, and education.

2. Prepare or Build Dataset for Training.
    a. Datasets:
        i. Kaggle's job listings dataset, OpenResume, or scraping job boards.
            1. https://www.kaggle.com/datasets/madhab/jobposts?utm_source=chatgpt.com
            2. https://www.kaggle.com/datasets/moyukhbiswas/job-postings-dataset?utm_source=chatgpt.com
            3. https://www.open-resume.com/?utm_source=chatgpt.com
            4. https://github.com/xitanggg/open-resume?utm_source=chatgpt.com
            5. https://www.kaggle.com/datasets/ravindrasinghrana/job-description-dataset?utm_source=chatgpt.com
    b. Preprocess the Data.
        i. Normalize text by removing unnecessary formatting.
        ii. Tokenize data for NLP tasks.
        iii. Annotate datasets for relevant features
    c. Split Data (20/80)

3. **Develop Skill Matching System**. Create an NLP-based system that identifies skill gaps between the job description and the user's skills.
    a. **Extract Key Information**:
        i. Use NLP libraries (e.g., spaCy, NLTK) to identify and extract job requirements and skills from job descriptions.
        ii. Process CVs to extract skills, work experience, and qualifications.
    b. **Match Skills**:
        i. Use similarity measures (e.g., cosine similarity, BERT embeddings) to compare job description skills and user skills.
        ii. Highlight missing or unmatched skills.
    c. **Score the Match**:
        i. Generate a matching score (0-100%) for CV-job alignment.
    d. **Integrate Feedback**:
        i. Provide suggestions to improve the match (e.g., add specific skills to the CV, adjust keywords).

4. **Create Content Generation System.** Automatically generate personalized resumes and cover letters tailored to the job description (Can be adjusted steps, cos I use CHAT4o to generate this section).
    a. Fine-Tune Generative AI Models:
        i. Use GPT-4 or similar models fine-tuned with your datasets for resume and cover letter generation.
    b. Resume Generation:
        i. Rearrange user-provided CV content to emphasize relevant experience and skills.
        ii. Use a template-driven approach for formatting.
    c. Cover Letter Generation:
        i. Generate tailored cover letters using user-provided descriptions and job information.
        ii. Structure user strengths and align them with the job description.
    d. Iterate:
        i. Incorporate feedback from the skill matching and feedback agent for further refinements.

5. **Feedback and Refinement System.** Build a feedback loop to improve generated documents.
    a. Integrate a Feedback Agent:
        i. Use rule-based or machine learning models to evaluate document quality.
    b. Score the Output:

        i. Provide a score (0-100%) for the generated CV and cover letter. Genereted cv or ect has to go through the **test**.
    c. **Suggestions for Improvement (Optional, can generate already improved version of cv):**
        i. Suggest adding specific skills, rephrasing sentences, or restructuring the format.
    d. **User Iteration:**
        i. Allow users to incorporate feedback and regenerate documents.

6. **Develop the User Interface.(Under discussion).**
    a. If we have time it could be HTML page with frontend if not can be through terminal*
7. **Environment.**
    a. Clone GitHub Link : https://github.com/Yehormann/NLPFinalProject.git
    b. Deploy project on **Master** branch only not **Main**.

Reference to Read before begin:

- https://towardsdatascience.com/a-comprehensive-guide-to-collaborative-ai-agents-in-practice-1f4048947d9c
- https://www.deeplearning.ai/short-courses/multi-ai-agent-systems-with-crewai/