ECE1786H Final Report

ResuMatch: An Agent-based Job Searching Tool

Submitted By: ResuMatch

Bart Cui, Karanbir Singh Brar, Lucas Rea, Rafay Kalim

Supervisor: Jonathan Rose

TA: Zafar Mahmood

Word Count: 1985

Introduction

ResuMatch is a job-searching tool designed to help candidates save time by providing precise job matches that align with their resume. The tool extracts key features and scans job postings by analyzing resumes to identify the best matches for the user's qualifications. ResuMatch aims to simplify job searching by helping candidates efficiently find roles that match their unique skills amid the overwhelming number of job listings. To achieve this, ResuMatch employs an agentic system which is an effective approach for tasks requiring content understanding, contextual processing, and decision-making.

System Illustration

Here we present the overall architecture of ResuMatch (Figure 1). The inputs to the system are the resume and the job location preference given by the user. The output of the system are good fit job postings in the form of URLs.

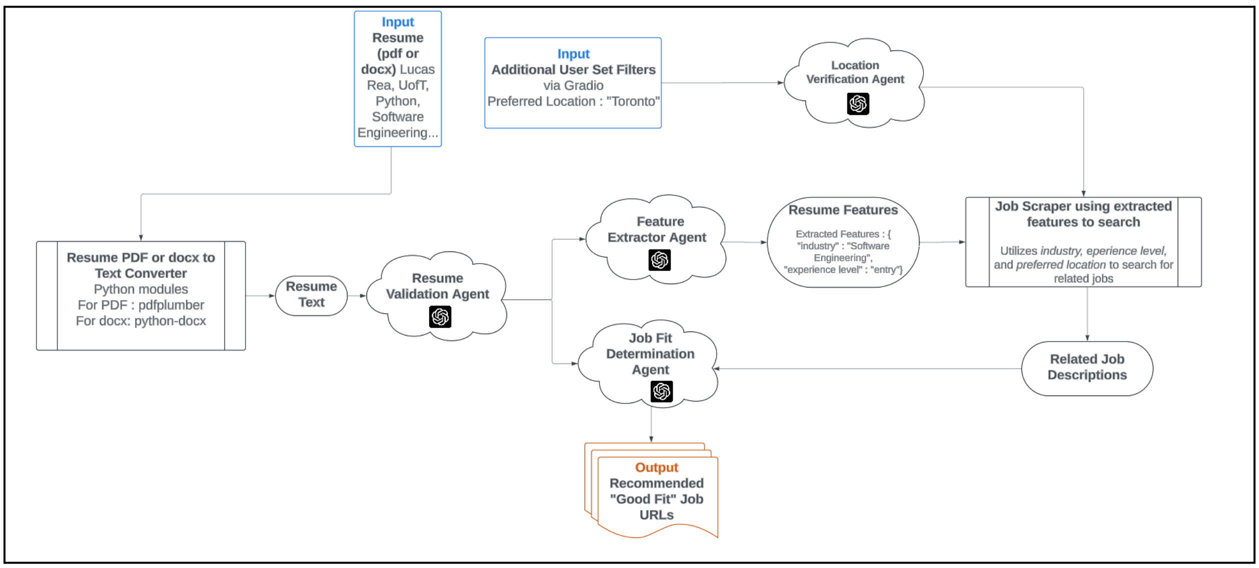


Figure 1: ResuMatch System

We've done work to improve the final output agent of the system - the Job Fit Determination Agent, as seen expanded in Figure 2. We use GPT-4o to summarize resumes and job descriptions in JSON format, mimicking a human recruiter's approach of categorizing key aspects to assess a candidate's fit for a role.

Figure 2: Job Fit Determination Agent

Background & Related Work

Hirize provides recruiters with an ML-based algorithm that efficiently screens large volumes of resumes for specific job postings [1]. Hirize parses resumes and transforms unstructured data into searchable fields that can be compared against job postings, enhancing its matching capabilities by employing company and industry mapping, hence categorizing resumes based on the candidate's background and job's requirements.

Another approach is “Resume Matching Framework via Ranking and Sorting Using NLP and Deep Learning" by Kumar [2] which employs NLP and models like BERT and GPT-3. The framework extracts key features and ranks resumes based on their relevance to specific job postings. While Hirize and Kumar’s framework focuses on ranking and sorting, ResuMatch takes a different approach by using a multi-agent system for more specialized processing of resumes and job postings.

Baseline Model / Comparison

Since we’ve built a class 2 project, our main method of evaluation is using datasets to evaluate the output for each agent. We built each agent class with an evaluation function which will take in a pandas dataframe via one of the CSV datasets. This makes it easy to run any validation test on our agents. We evaluate our agents on the percentage of fields in their respective testing datasets they predict correctly.

Data and Data Processing

The primary purpose of data for our project is the evaluation of our agents. We have one dataset for each agent. First, for the Feature Extraction Agent, we used a dataset from HuggingFace containing resume text, and a corresponding industry [3]. The data cleaning performed was to balance the class labels and reorganize some niche industry names into more comprehensive titles. Figure 3 shows that our preprocessing was successful and that the industry label is distributed more equally than originally. We manually assigned an experience level from entry, junior, senior and executive to each row since these are the two fields our feature agent will fetch.

Industry	resume_text	experience_level
Architecture	jessica claire montgomery street san francisco...	senior
Accountant	jessica claire 100 montgomery st 10th floor 55...	junior
Testing	manual qa tester cv sample philip jennings 122...	entry

Table 1: Examples from ResumeFeatures.csv

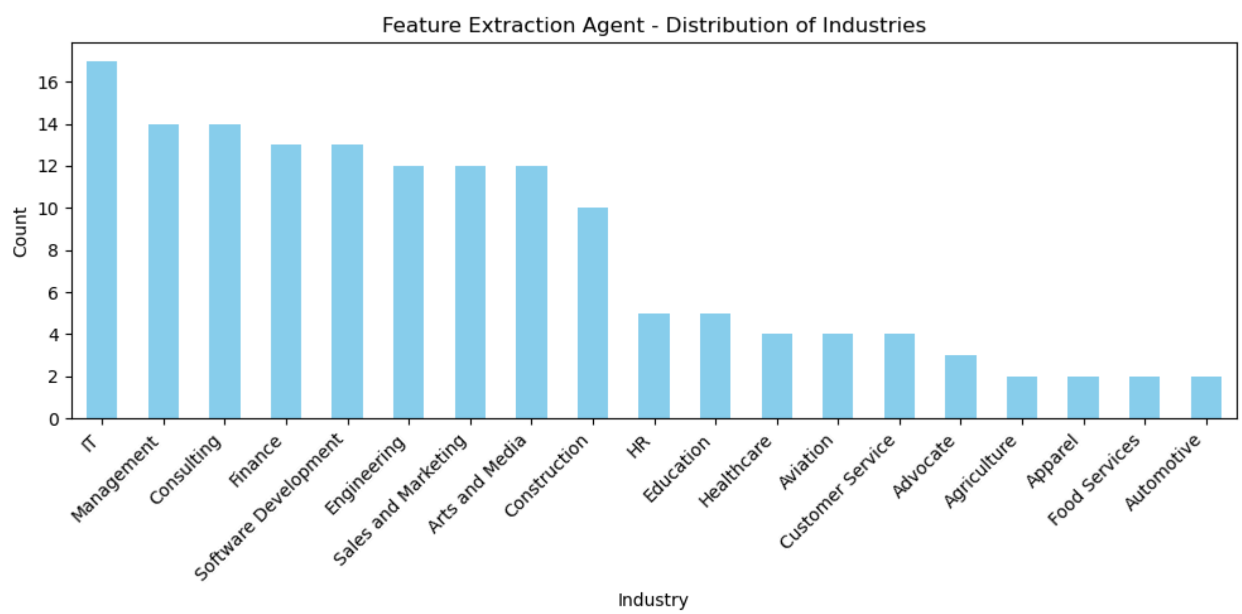


Figure 3: Distribution of Industries in Resume Dataset

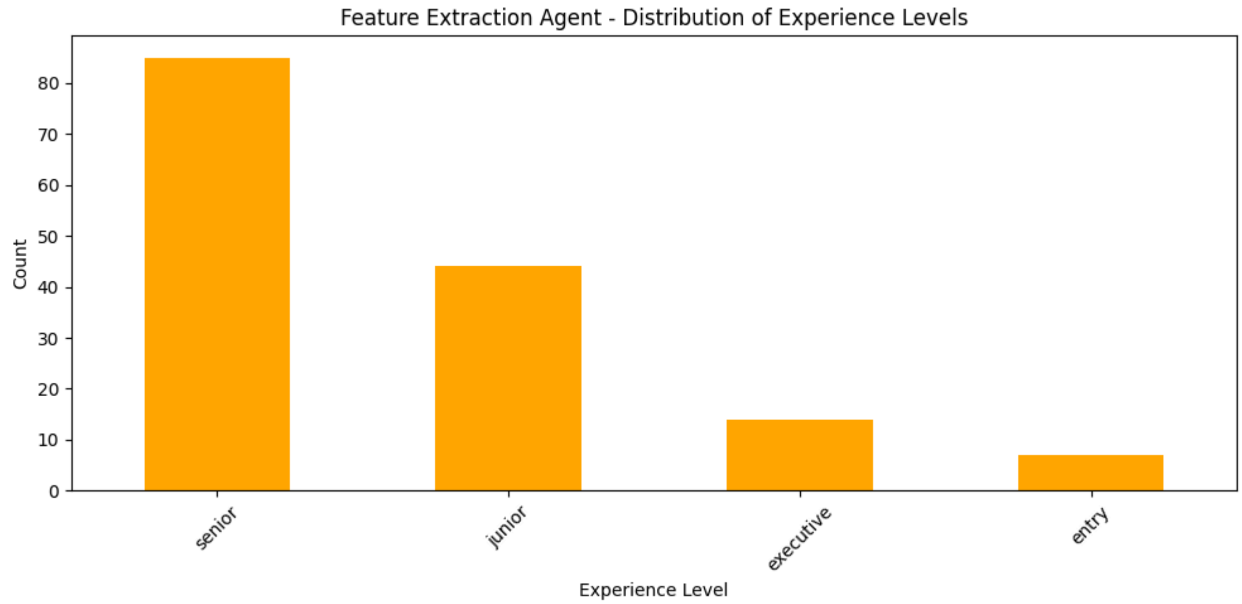


Figure 4: Distribution of Experience Levels in Resume Dataset

Our second dataset, also from HuggingFace, is for the Job Fit Determination Agent [4]. This dataset has three columns: resume text, job description text, and a labelled field of "Good", "Potential", and "No" fit. We sampled an equal amount of data from each fit label to ensure the evaluation dataset was balanced, which can be seen in Figure 5.

resume_text	job_description_text	label
SummaryEnergetic worker with a broad range of ...	Reviews, analyzes, and evaluates business syst...	Good Fit
Professional Overview10+ years' research exper...	Hello,Greetings from DevCare SolutionsI got an...	No Fit
ProfileHighly motivated Sales Associate with e...	Are you passionate about the clean tech automo...	Potential Fit

Table 2: JobResumeMatches.csv - Evaluation of Job Fit Determination Agent

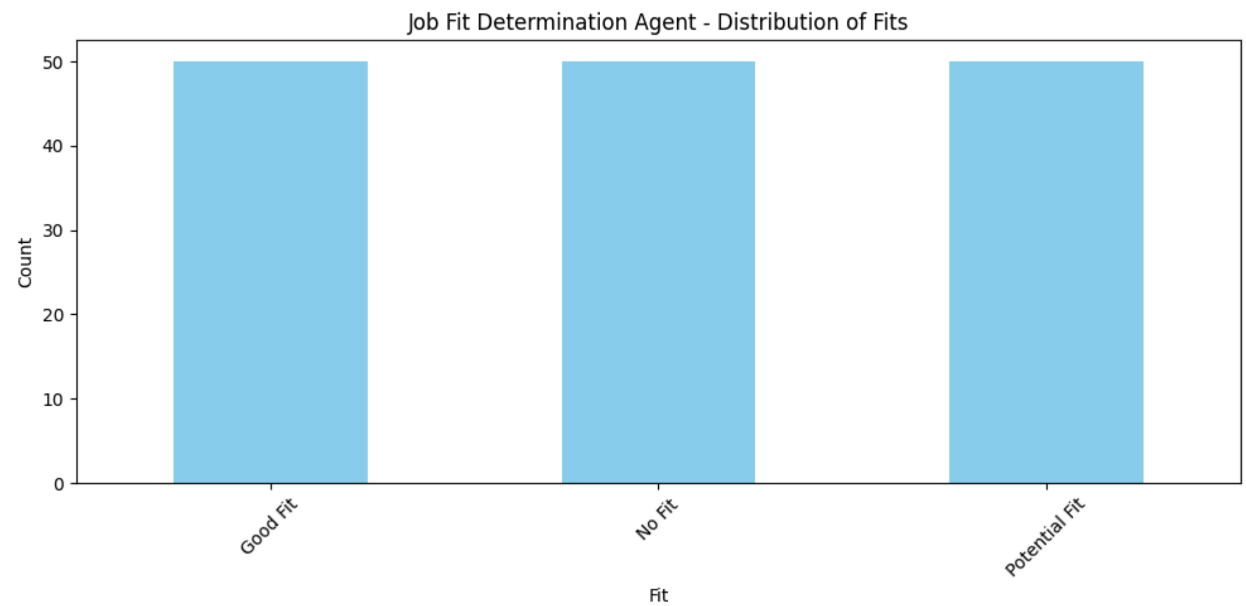


Figure 5: Distribution of Fit Labels in Job Resume Matching Dataset

Our third and fourth dataset were manually created and used for the evaluation of the Location Validation Agent and Resume Validation Agent. For the Location Validation Agent, we created a dataset with 25 rows of locations from around the world, with label “1”, and 25 randomly generated locations with label “0”. For the Resume Validation Agent, we took 25 resume text rows from our Job Fit Determination Agent dataset with the label “1”, and 25 irrelevant movie reviews from the IMDB Review Dataset with the label “0”.

Location Validation Dataset Sample	**Resume Validation Dataset Sample**
Text / Label Bay Area / 1 Unicornville / 0	resume_text / label education omba executive leadership university... / 1 I really liked this Summerslam due to the look... / 0

Table 3: Verification Agent Sample

Architecture and Software

The agentic system we’ve implemented has numerous components, utilizing GPT-4o agents and other forms of Python automation.

Architecture

User Input

The Gradio UI enables users to upload their resume in PDF or DOCX format and specify preferred job locations.

Feature and Validation Agents

As part of the resume processing, we employ a GPT-4o agent to validate if the processed text is a resume. This ensures the system does not run on phony text data.

We use a GPT-4o agent to validate user location, which accommodates flexible inputs like "Bay Area" or "Pacific Northwest" which traditional location validators might reject.

After this validation is complete and verified, the resume text is passed to the Feature Extraction agent. This agent is responsible for parsing key information from the resume – the industry of the candidate, as well as their level of experience.

Job Scraping

Using the features extracted from the Feature Extraction agent, along with the validated location we're able to scrape various job boards for related job postings – returning the job details.

Job Fit Determination

Once the Job Scraper returns relevant jobs, the job description text and with the previously parsed resume text are passed to the Job Fit Determination Agent. This agent is responsible for determining which jobs are the best fit for the candidate's resume.

The agent uses GPT-4o to transform unstructured resume and job description data into structured JSON. This approach mimics a recruiter's process of systematically analyzing key features and requirements to determine job fit, extracting and categorizing crucial information for precise matching.

```
{
    "Summary": "Energetic worker with a broad range of customer service an
    "Highlights": [
      "Certified in 10-key",
      "Time management",
      "Meticulous attention to detail",
      "Results-oriented"
    ],
    "Education": {
      "Institution": "Florida State College at Jacksonville",
      "Location": "Jacksonville, Florida",
      "Expected Graduation": "2014",
      "Degree": "Associate of Arts in Business Administration",
      "GPA": "3.60",
      "Honors": [⋯
      ],
      "Coursework": [⋯
      ]
    },
    "Experience": [
      {
        "Company": "Iron Mountain Incorporated",
        "Position": "Data Entry Operator",
        "Location": "Essex Junction, VT",
        "Duration": "08/2009 - Current",
        "Responsibilities": [⋯
        ]
      },
```


Figure 6 : Sample JSON summary of resume


```
{
  "JobDescription": {
    "Responsibilities": [
      "Review, analyze, and evaluate business systems and user needs",
      "Formulate systems to align with business strategies",
      "Business process reengineering",
      "Identify new technology applications",
      "Prepare solution options, risk identification, and financial analyses",
      "Write detailed descriptions of user needs and program functions",
      "Work on Grants Management application",
      "Perform duties of a Business Analyst",
      "Coordinate with technical teams for demos, UAT, code releases, and user communication",
      "Work in an Agile environment",
      "Provide on-call support as needed"
    ],
    "SkillsAndQualifications": {
      "MinimumRequirements": {
        "BusinessAnalystExperience": "8 years",
        "SQLExperience": "8 years",
        "SystemReporting": "8 years",
        "SystemsTesting": "8 years",
        "CommunicationSkills": "Excellent"
      },
      "Preferred": {
        "AgileExperience": "4 years",
        "ProjectOrProductManagement": "2 years",
        "JiraExperience": "1 year"
      }
```


Figure 7: Sample JSON Summary of job

The final call to GPT-4o then determines which jobs are the best fit for the candidate's resume. This is done with a prompt that describes what a "Good Fit", "Potential Fit", and "No Fit" are in the context of ResuMatch.

Good fit jobs are returned to the Gradio UI as clickable URLs for the candidate to explore.

Quantitative Results

Feature extraction agent in ResuMatch is evaluated by using two confusion matrices to understand the agent's accuracy across industry categories and experience levels. The agent's baseline accuracy of 17.33% across 35 industries. Another GPT-4o mini model was used to further classify industries down to 20 categories. With additional prompt engineering, the agent's accuracy improved to 58% for feature extraction and 74% across the four experience levels. For example, the agent was best at distinguishing finance roles. However, the agent did misclassify software development and IT roles, indicating areas for further agent improvement.

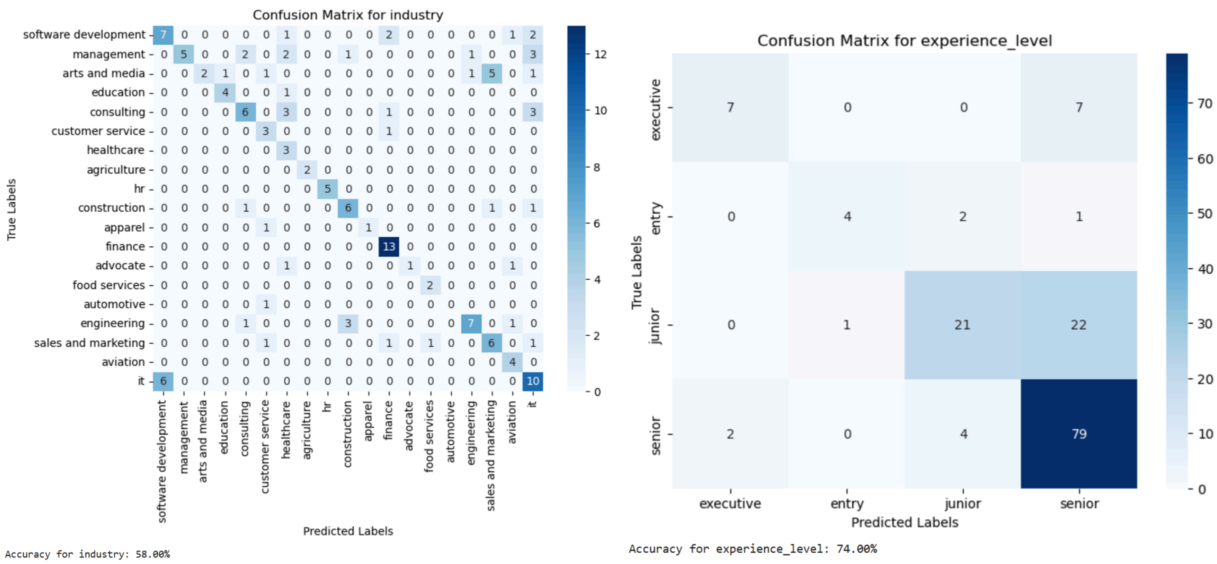


Figure 8: Confusion matrices for feature extraction agent

As for the resume and location validation agents, it seems that is an easier task for GPT-4o to perform. It can get 100% accuracy for both location and resume validation. Our prompts have basic instructions and examples for this task, and it is able to do well.

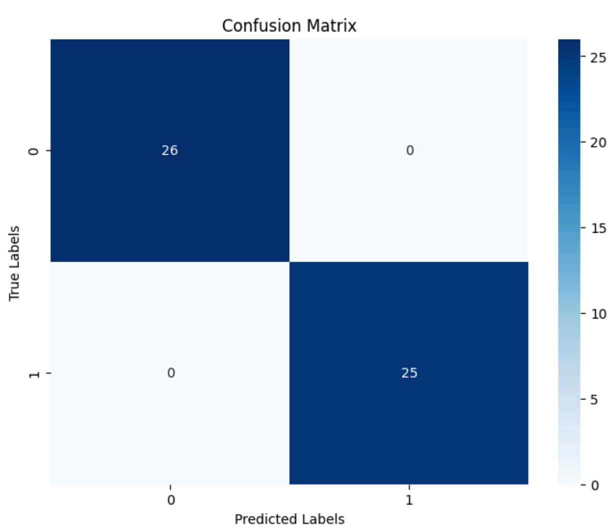
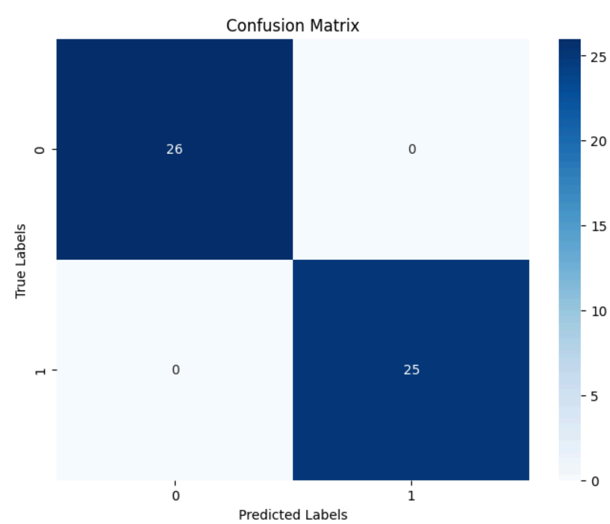
Agent	Resume Validation	Location Validation
Accuracy	100%	100%
Confusion Matrix		

Table 4: Resume and Location Validation Results

For evaluation of the final agent, the Job Fit Determination agent, we utilized the dataset that contains pairs of resumes and job descriptions along with a label of “Good Fit”, “Potential Fit”, or “No Fit”.

The final confusion matrix after experimenting with various prompts and agent processes is as follows – with an overall accuracy of 48% (baseline agent was 44.67%).

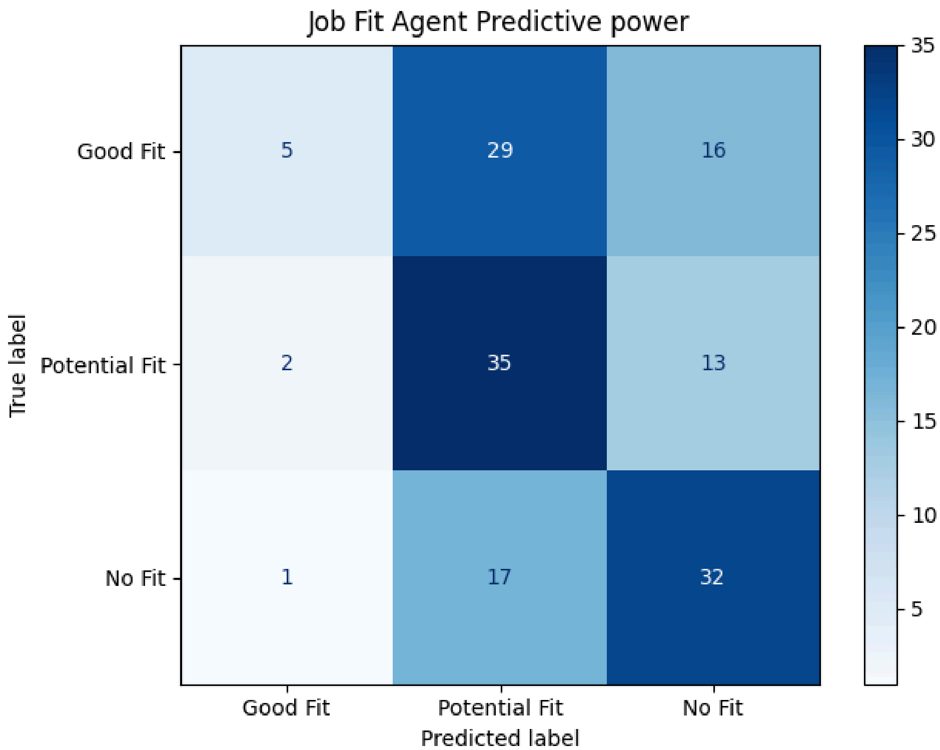


Figure 9: Confusion matrix for Job Fit Determination

These results demonstrate an improvement over the baseline results reported in our progress report, indicating that the system has been successful in meeting its objectives. For experimentation results, see Appendix.

Qualitative Results

Job Matching Quality

The system's performance varies depending on the industry, though is still seen as a success. In data science and machine learning resumes, where education and work experience are more aligned, the system performs optimally. However, for resumes like an accountant, with disparate work experiences such as part-time roles that do not align with educational background, the results can be less coherent. This is primarily because the system does not evaluate the chronological progression of a candidate's professional and educational history.

Qualitative results are shown in Table 5.

Resume	jobs returned
Accounting	Clerk Seasonal Package Handler-Warehouse Clerk Part-Time Cashier / Line Cook
Administrative assistant	Event Representative Entry-Level Office Assistant
data science and machine learning	Junior Data Analytics Developer Junior Data Analyst/Engineer/Scientist - Remote Data Scientist - Machine Learning, NEW GRADS WELCOME (Calgary, AB) Junior Data Analytics Developer Ubuntu Sales Engineer (Entry-Level) Data Scientist - Machine Learning Junior Data Scientist and ML Engineer
Finance	Entry-Level Finance Analyst - Eastern Canada District - 2025
human resources	Entry Level Customer Service Assistant

Table 5: ResuMatch results – varying industry

System Robustness

The resume validation agent confirms that the input file is a resume. An example of the working agent is provided below. For this, we demonstrate the ability of the system to distinguish between resumes and other documents Figure 10.

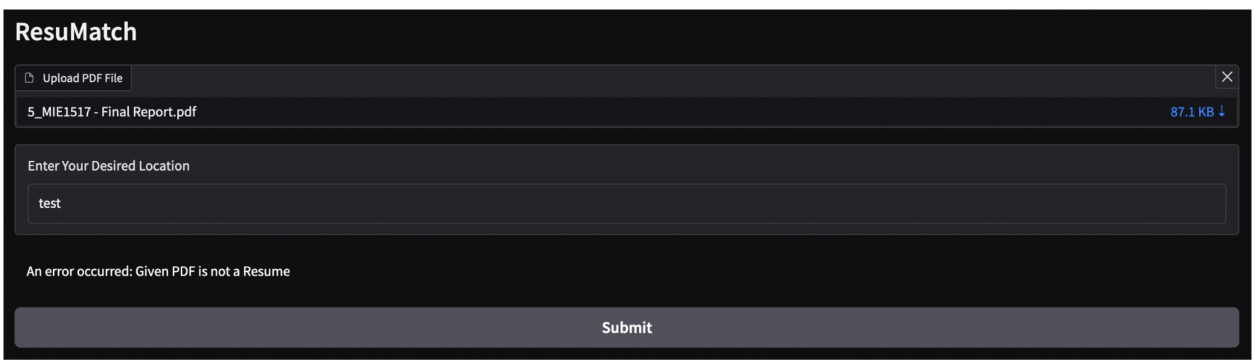


Figure 10: Handling invalid file

The location agent using GPT-4o module gives an advantage of taking as input misspelled locations and returning results for the closest location matching the input as illustrated in Figure 11. If the entered location is not spelled similarly to any valid location, the system returns an error as shown in Figure 12.

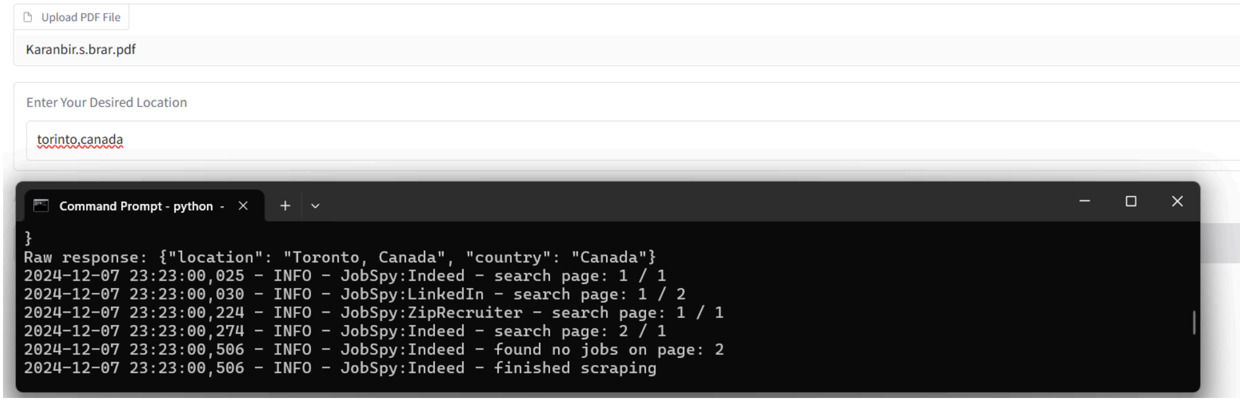


Figure 11: System functions with misspelled location

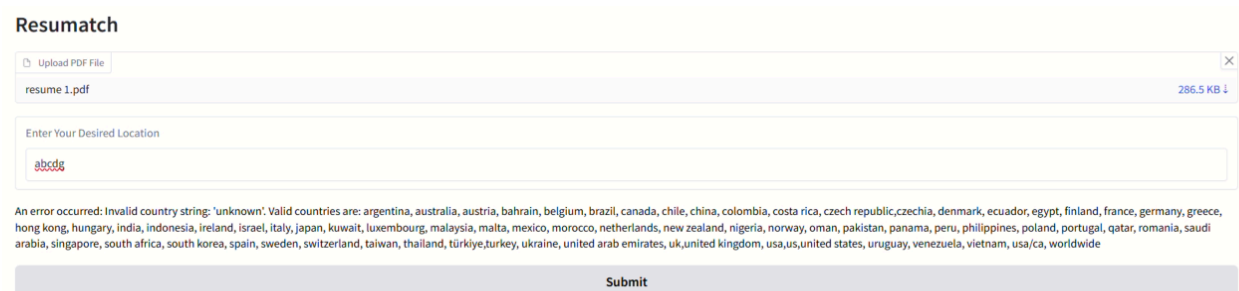


Figure 12: Completely invalid location

Discussion and Learnings

We are quite happy with the performance of all agents and the system as a whole. As mentioned in the Quantitative Results section, our agents perform strongly across the datasets and our prompt engineering efforts were fruitful.

One thing we would have liked to have known before we started was how we were going to use data for this project. We had collected a lot of relevant data at the start but without a purpose for it, which we figured out halfway through the project.

Secondly, one interesting discussion point is that our problem and data is quite subjective. While we have treated the datasets as “ground truth”, we must acknowledge that there is some bias in us labelling experience levels of resumes ourselves, as we are not experienced professionals in the respective fields of the resumes. Thus, the performance of our agents may be better or worse as we refine the quality of the dataset itself.

Lastly, we note that our agents currently do not take into account the chronological ordering of the resume. For example, it may not identify career switchers and still suggest jobs related to a user’s previous career rather than their new one.

Individual Contributions

Bart

- In the resume-parser branch, developed the parser and feature extraction agent.
- Integrated individual parts of the parser and feature extraction agent into the main project after the progress report.
- In the resume-phase-2 branch, implemented an LLM to further classify industries.
- Fine-tuned the feature extraction agent to improve its accuracy.
- Added test execution requirements and procedures to the README.

Lucas

- Found job matching and resume feature datasets
- Labeled 40 rows of the “Experience Level” column for feature extraction.
- Performed data cleaning on resume feature dataset
- Developed agent architecture and validation code, multi-step agent for job fit
    - Experiment and perform prompt engineering for better final agent results
- Developed job matching agent
- Enhance README, code quality

Rafay

- Created Gradio UI and integrated it with the system.
- Downloaded, preprocessed datasets, and streamlined their integration into workflows.
- Developed and evaluated the feature extraction agent.
- Manually labeled and created datasets for resume and location validation.
- Prompt-engineered and created the resume validation agent.
- Labelled 40 rows of the “Experience Level” column.
- Enhanced README, code quality, and software architecture.
- Implemented secure method for OpenAI API key usage without uploading to GitHub.
- Integrated the Resume Validation Agent with the system.

Karanbir

- Incorporated a job scraper to download jobs by scraping 5 websites based on features extracted from the resume.
- Created a location agent to assist with location entry and prediction.
- Verified the performance of the location agent and refined its prompt through prompt engineering.
- Compared two approaches for the location agent: Using the Geolocation module and using the GPT-4o module.
- Integrated all agents with the UI to form the final product.

References

[1] “How Hirize’s Machine Learning-Based Job Resume Matching Works (part I),” Hirize. Accessed: Nov. 03, 2024. [Online]. Available: https://hirize.hr/blogs/hirize-machine-learningjob-resume-matching

[2] S. Tanberk, S. S. Helli, E. Kesim, and S. N. Cavsak, “Resume Matching Framework via Ranking and Sorting Using NLP and Deep Learning,” in 2023 8th International Conference on Computer Science and Engineering (UBMK), Sep. 2023, pp. 453–458. doi: 10.1109/UBMK59864.2023.10286605.

[3] “ahmedheakl/resume-atlas · Datasets at Hugging Face.” Accessed: Nov. 03, 2024. [Online]. Available: https://huggingface.co/datasets/ahmedheakl/resume-atlas USED

[4] “cnamuangtoun/resume-job-description-fit · Datasets at Hugging Face.” Accessed: Nov. 03, 2024. [Online]. Available: https://huggingface.co/datasets/cnamuangtoun/resume-job-description-fit

Permission:

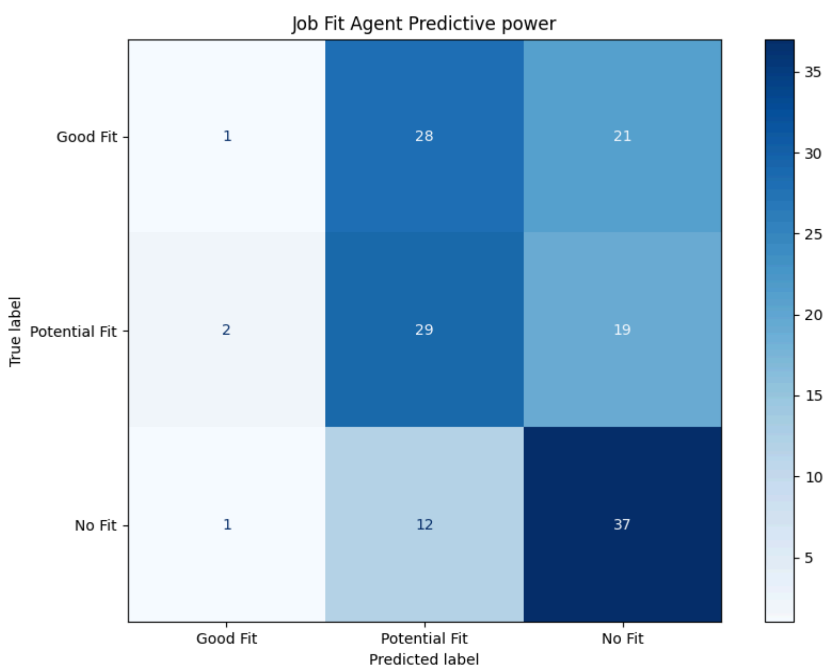
	Permission to Post Video	Permission to Post Final Report	Permission to Post Source Code
Rafay	Yes	Yes	Yes
Bart	Yes	Yes	Yes
Lucas	Yes	Yes	Yes
Karanbir	Yes	Yes	Yes

Appendix

Job Fit Determination Experimentation

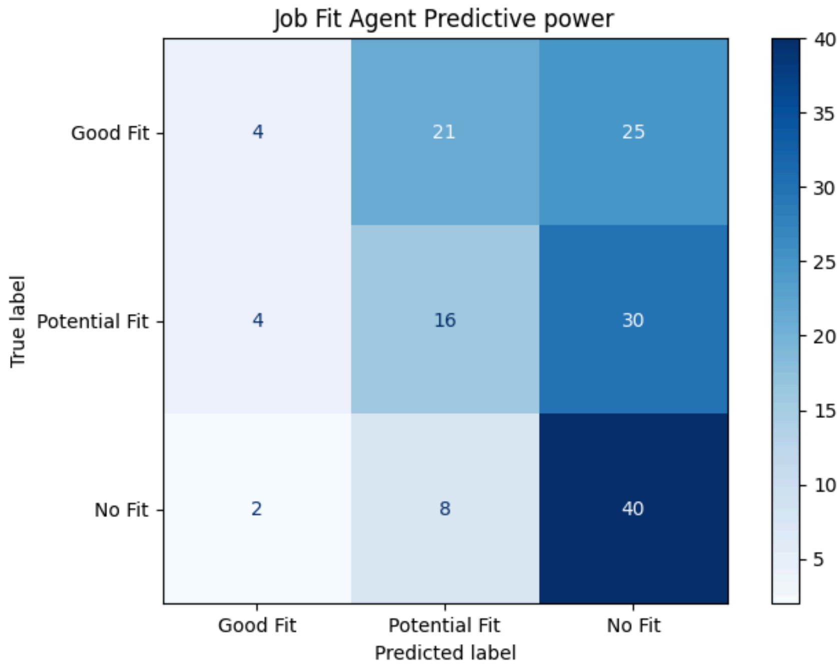
BASELINE

Success rate: 44.67%



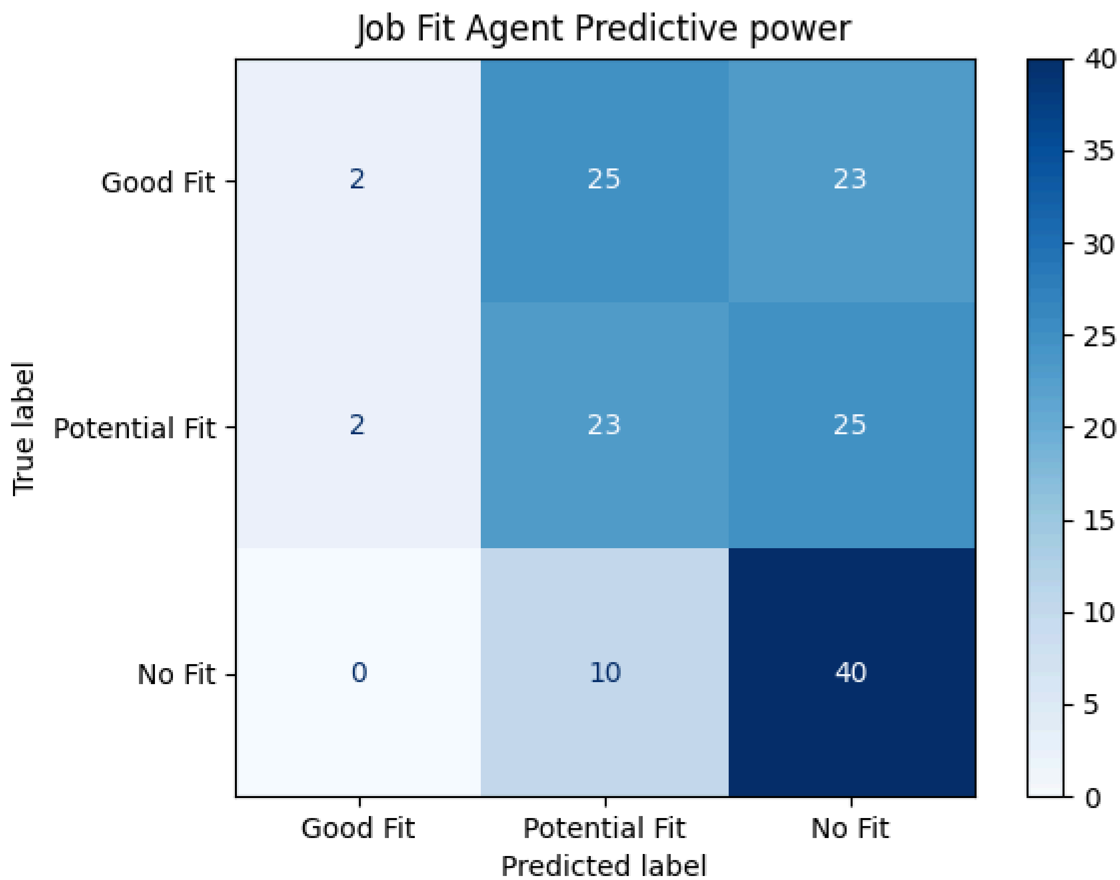
Test 1) Try 4o-mini instead of 4o

Success rate: 40.00%



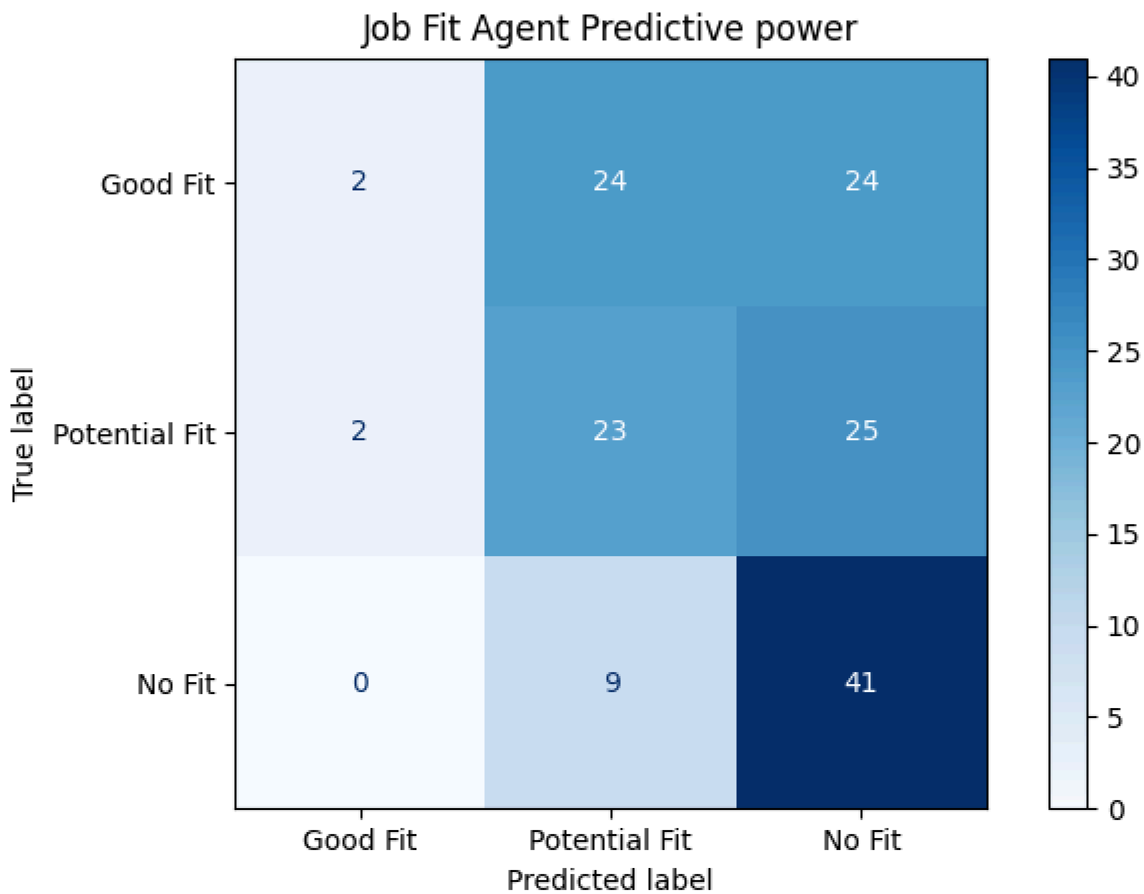
Test 2) 4o. Classify what good, potential, and no fits are.

Success rate: 43.33%



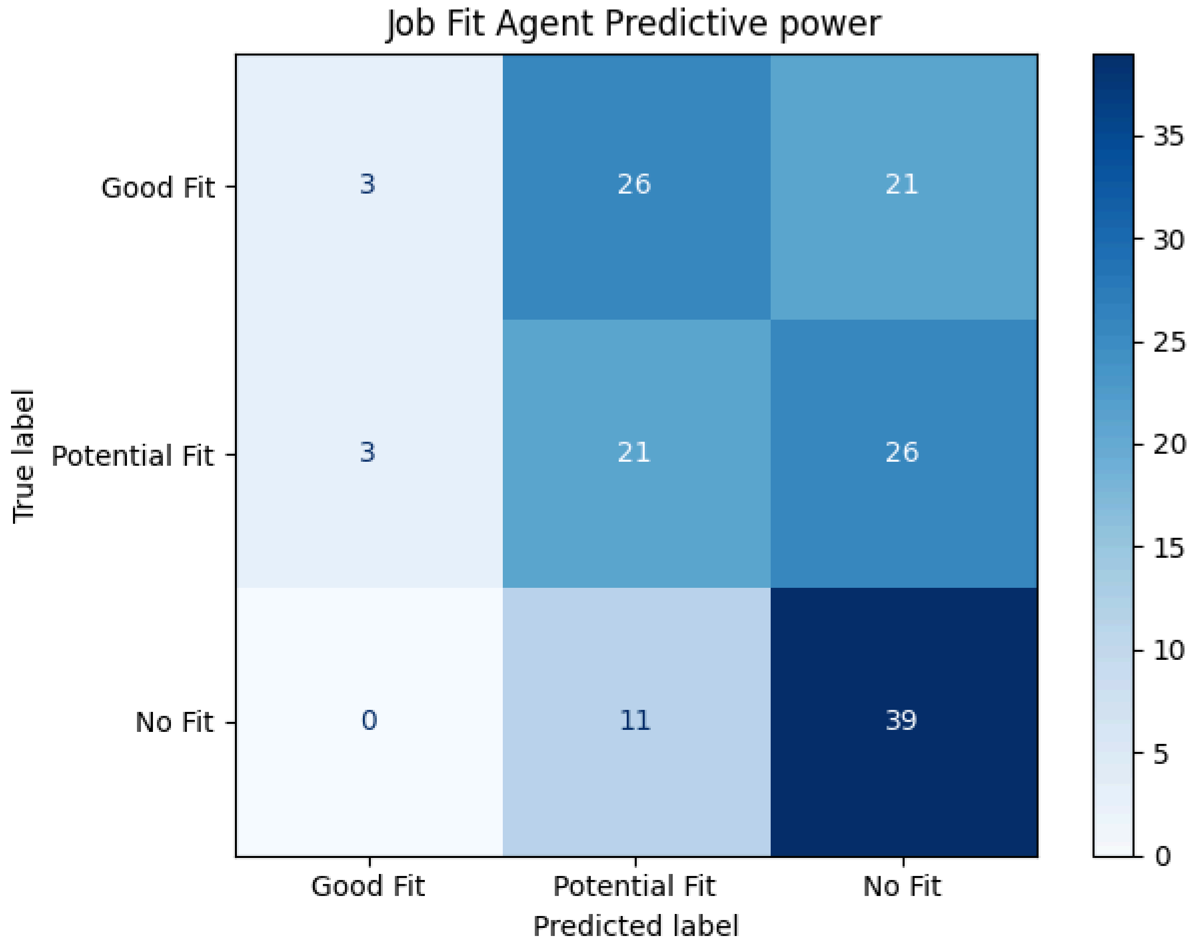
Test 3) Same as test 2), temp=0.5

Success rate: 44.00%



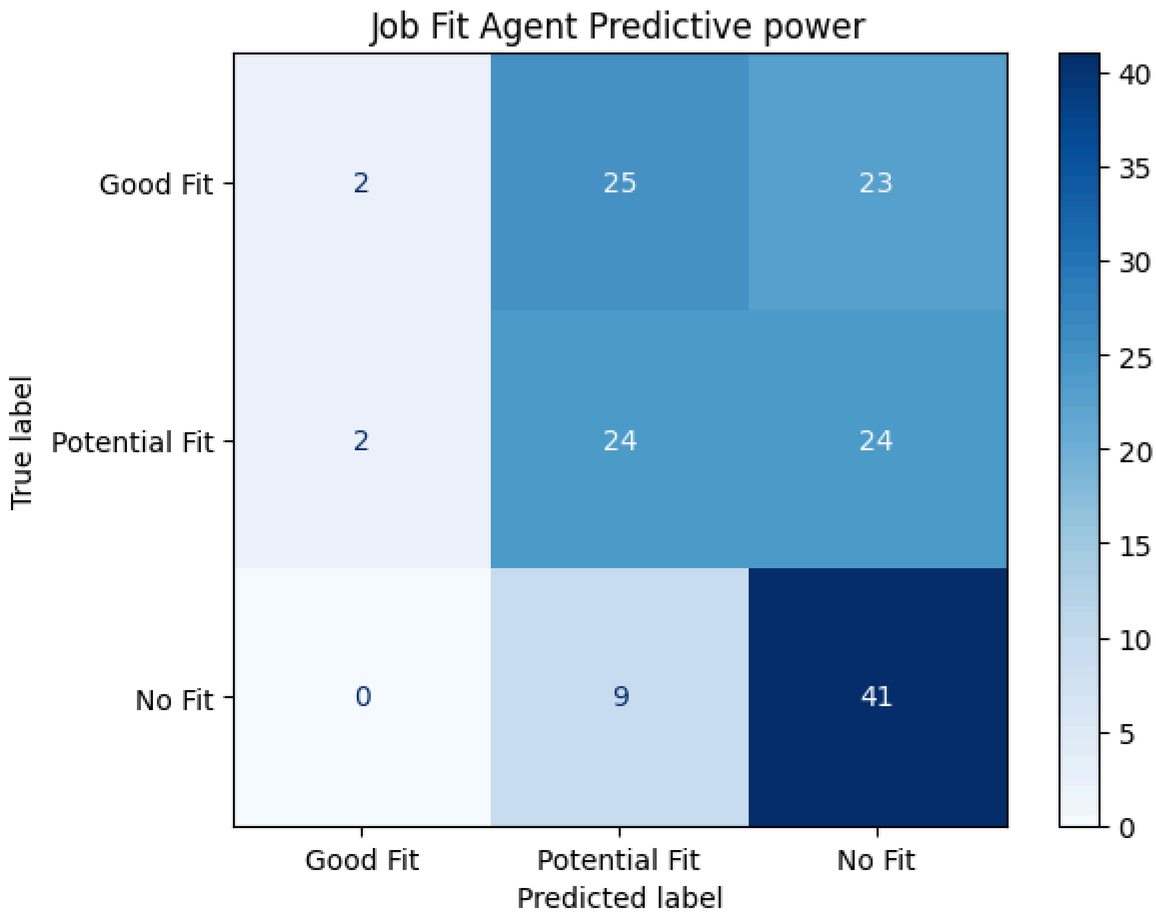
Test 4) Same as test 3), temp=0.85

Success rate: 42.00%



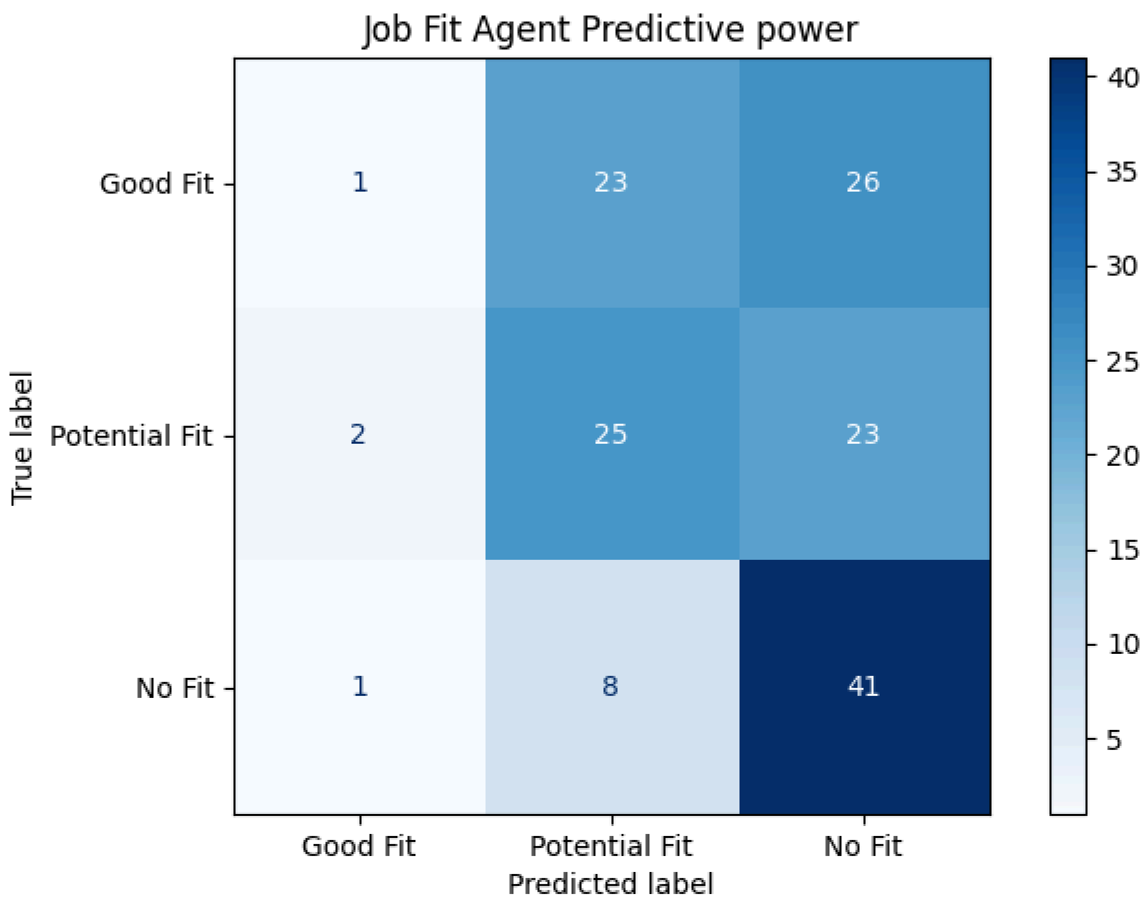
Test 5) Same as test 4), temp=0.15

Success rate: 44.67%



Test 6) Mulit-step agent process 1000 token JSON summary

The process here was to make 2 requests to gpt-4o. To get a brief summary of the resume text, as well as job description. This would summarize the long amount of text to hopefully the most important points. Then this info would feed to the actual matching request.
Success rate: 44.67%

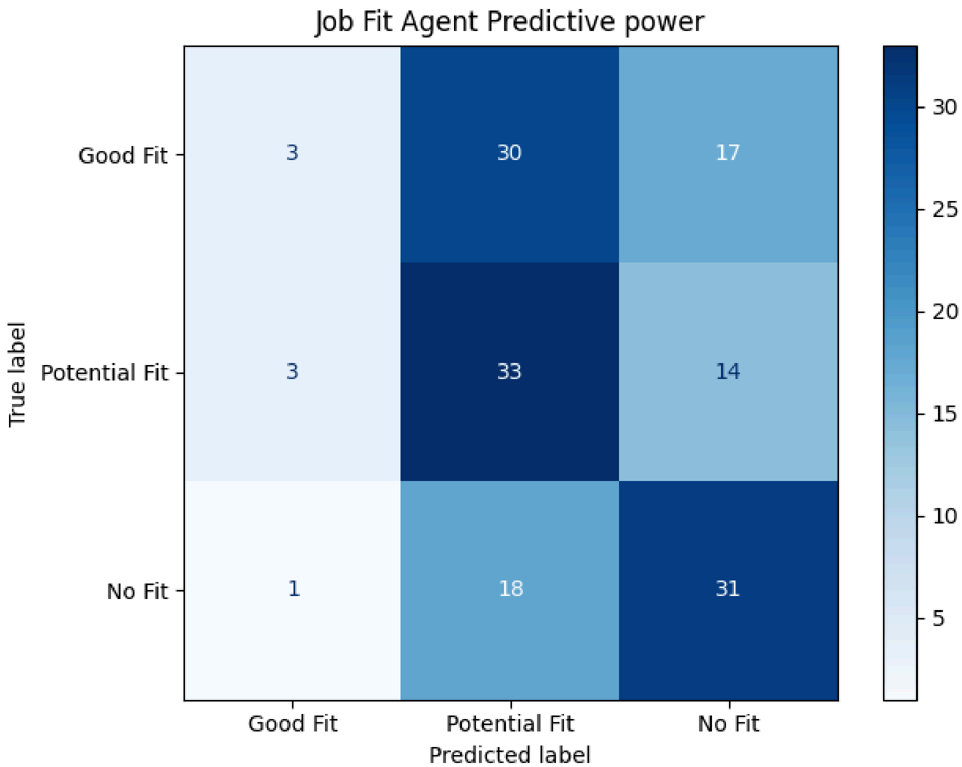


Test 7) Multi-step agent with best prompts

Success rate: 44.67%
Can interpret this result to actually be pretty good. If a job is a good fit, it's also a potential fit. Perhaps the data labels are somewhat not aligned properly. This result is promising because most correct "good fits" are in the good and potential category (3+30 / 50).

Similarly, if something is definitely a no fit, it may very well be a potential fit (once again depends on all of these data labels. Though the majority are assigned in no fit.

An overall accuracy score may also not be the best for this. Since we may want to give a 'half point' for a good fit being predicted as a potential fit.


Job Fit Agent Predictive power
True label
Good Fit
Potential Fit
No Fit
3
30
17
3
33
14
1
18
31
Good Fit
Potential Fit
No Fit
Predicted label
30
25
20
15
10
5

Test 8) Same as test 7), allowing more tokens (2500) in JSON summary

Success Rate: 48.00 %