INTELLIGENT JOB RECOMMENDATION ENGINE

Phase 1 Report: Data Collection & Resume Parser Implementation

Member 1 : Ashfaq Ali Anees Ali 40406560
Member 2 : Naveen Kumar Raju 87474061
University of Europe for Applied Sciences
Date: 31-10-2025

---

1. Introduction

1.1 Project Overview

Candidates now face a more complicated job search procedure due to the abundance of internet job ads. Job searchers often waste time applying for positions that are not a good fit for their expertise and skill set. Similarly, recruiters have trouble matching candidates to job requirements.

The goal of this project is to create an intelligent job recommendation system that automatically matches job seekers with qualified employment opportunities using machine learning and natural language processing. The technology creates customized job recommendations by analyzing resumes to extract candidate abilities and comparing them to job criteria.

1.2 Phase 1 Objectives

The primary objectives of Phase 1 were:

- Gather suitable data sets using information from job postings and resumes.
- Examine and comprehend the datasets' content and structure.
- To guarantee quality, clean and preprocess the data.
- Create a resume parser that can identify abilities in resume content.
- Verify and test the accuracy of the parser.

---

2. Data Collection

2.1 Datasets Used

Three datasets were collected from Kaggle to support this project:

Dataset 1: Resume Dataset

- **Source:** https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset
- **Total Records:** 2,484 resumes
- **Columns:** ID, Resume_str (text), Resume_html (HTML format), Category
- **Categories:** Resumes span multiple professional categories including HR, IT, Teaching, Healthcare, Design, and others
- **Purpose:** Primary dataset for testing skill extraction capabilities

Dataset 2: LinkedIn Job Postings

- **Source:** https://www.kaggle.com/datasets/arshkon/linkedin-job-postings
- **Total Records:** 123,849 job postings
- **Key Columns:** job_id, company_name, title, description, location, formatted_work_type, formatted_experience_level
- **Purpose:** Provides job descriptions and requirements for matching candidates

Dataset 3: Job Skills Mapping

- **Source:** https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset
- **Total Records:** 213,768 job-skill mappings
- **Columns:** job_id, skill_abr (skill abbreviation)
- **Unique Jobs:** 126,807
- **Unique Skills:** 35 skill categories
- **Purpose:** Maps jobs to required skill categories for Phase 2 matching

2.2 Dataset Selection Rationale

These datasets were selected because they provide:

- Real-world resume examples across various industries
- Large-scale job posting data with detailed descriptions
- Structured skill mappings for future ML model training
- Sufficient volume for meaningful analysis and testing

---

3. Data Exploration

3.1 Resume Dataset Analysis

Initial exploration of the resume dataset revealed:

Data Structure:

- No missing values detected
- Resume text stored in both plain text and HTML formats

Findings:

- Average Resume length: 811.32 words
- Shortest Resume: 0 words
- Longest Resume: 5190 words
- Most common category: INFORMATION-TECHNOLOGY (120 Resume)

Category Distribution:

INFORMATION-TECHNOLOGY	120
BUSINESS-DEVELOPMENT	120
ADVOCATE	118
CHEF	118
ENGINEERING	118
ACCOUNTANT	118
FINANCE	118
FITNESS	117
AVIATION	117
SALES	116
BANKING	115

HEALTHCARE	115
CONSULTANT	115
CONSTRUCTION	112
PUBLIC-RELATIONS	111
HR	110
DESIGNER	107
ARTS	103
TEACHER	102
APPAREL	97
DIGITAL-MEDIA	96
AGRICULTURE	63
AUTOMOBILE	36
BPO	22

Sample Resume Observation: Resumes contain typical sections including professional summary, work experience, education, and skills. The text format varies significantly, with some containing special characters, HTML tags, and inconsistent formatting.

3.2 Job Postings Dataset Analysis

Initial Statistics:

- Total job postings: 123,849
- Total columns: 31
- Companies represented: 24422
- Locations covered: Multiple countries and cities

Key Findings:

- Job descriptions differ greatly in length and level of detail.
- Numerous columns have large null value percentages.
- About 24 percent of posts lack experience level information.
- Only 24% of records provide complete salary information.

3.3 Job Skills Dataset Analysis

The job skills dataset provided valuable insights into skill requirements:

- 213,768 skill mappings in total
- Abbreviations are used to code skills (e.g., FIN for Finance, MRKT for Marketing).
- Certain jobs demand a variety of skills.
- Average abilities for each position: The ratio of 126,807 unique jobs to 213,768 mappings is 1.68.

---

4. Data Cleaning

4.1 Job Postings Data Cleaning

Step 1: Column Reduction

Many columns were not relevant for the job-candidate matching task. The following 22 columns were removed:

- views, applies (engagement metrics, not relevant for matching)
- original_listed_time, expiry, closed_time (temporal data, not needed)
- job_posting_url, application_url, application_type (URL information)
- pay_period, max_salary, min_salary, med_salary (incomplete salary data)
- company_id (redundant with company_name)
- remote_allowed (high null percentage)
- listed_time, posting_domain, sponsored (not relevant)
- currency, compensation_type, normalized_salary (incomplete)
- zip_code, fips (location detail not needed)

Retained Columns:

- job_id, company_name, title, description, location
- formatted_work_type, formatted_experience_level, work_type

Step 2: Handling Missing Values

Critical columns with missing values were addressed:

- **Description nulls:** 7 rows removed
- **Company name nulls:** 1,719 rows removed
- **Experience level nulls:** 29,409 values filled with **"Not Specified"**
- **skills_desc column:** Dropped entirely (**98% null values**)

Step 3: Text Normalization

- Converted all job descriptions to lowercase
- Removed leading and trailing whitespace
- Ensured consistent text format

Before and After Statistics:

Metric	Before Cleaning	After Cleaning
Total Rows	123849	122124
Total Columns	31	8

4.2 Resume Data Cleaning

The resume dataset required minimal cleaning:

- **Null values:** None detected
- **Duplicates:** None detected
- **Text processing:** Converted Resume_str to lowercase and stripped whitespace

Final resume dataset: 2484 clean records ready for parsing

4.3 Job Skills Data Cleaning

- **Null values:** None detected
- **Duplicates:** Removed duplicate job_id and skill_abr combinations
- **Final dataset:** 126807 unique job-skill mappings

---

5. Resume Parser Implementation

5.1 Approach and Methodology

For Phase 1, a keyword-based skill extraction approach was implemented. This method was chosen because:

- **Simpleness:** Easy to execute within the Phase 1 schedule
- **Effectiveness:** Good for obtaining common soft and technical skills
- **Transparency:** Debugging and understanding are simple
- **Baseline:** Offers a starting point for further stages of more complex NLP techniques.

Alternative approaches considered:

Named Entity Recognition (NER) using spaCy - more complex, reserved for **Phase 2**

5.2 Comprehensive Skill List Development

A comprehensive skill list was created covering multiple categories:

Programming Languages: Python, Java, JavaScript, C++, C#, PHP, Ruby, Swift, Kotlin, Go, Rust, TypeScript, Scala, R, MATLAB, Perl, Shell scripting

Web Development: HTML, CSS, React, Angular, Vue, Node.js, Express, Django, Flask, Spring Boot, ASP.NET, jQuery, Bootstrap, Sass, Webpack

Databases: SQL, MySQL, PostgreSQL, MongoDB, Oracle, SQL Server, SQLite, Redis, Cassandra, DynamoDB, Firebase, NoSQL

Cloud & DevOps: AWS, Azure, Google Cloud, Docker, Kubernetes, Jenkins, Git, GitHub, GitLab, CI/CD, Terraform, Ansible, Linux, Unix

Data Science & Machine Learning: Machine Learning, Deep Learning, Data Analysis, Data Science, Artificial Intelligence, NLP, Computer Vision,

TensorFlow, PyTorch, Keras, Scikit-learn, Pandas, NumPy, Matplotlib, Tableau, Power BI

Mobile Development: Android, iOS, React Native, Flutter, Xamarin

Testing & QA: Selenium, JUnit, Pytest, Testing, Quality Assurance, Automation Testing

Project Management: Agile, Scrum, Kanban, JIRA, Waterfall, Project Management

Soft Skills: Communication, Leadership, Teamwork, Problem Solving, Analytical, Critical Thinking, Time Management, Collaboration

Business Tools: Excel, PowerPoint, Word, Microsoft Office, Google Docs, Outlook

Other Technical Skills: REST API, GraphQL, Microservices, API, JSON, XML, SAP, Salesforce, Cybersecurity, Networking, Blockchain, IoT

Total Skills in List: 117

5.3 Parser Implementation

The resume parser was implemented as a Python function using regular expressions for pattern matching:

```
def extract_skills(resume_text, skills_list):
    resume_text = resume_text.lower()
    found_skills = set()

    for skill in skills_list:
        pattern = r'\b' + re.escape(skill) + r'\b'
        if re.search(pattern, resume_text):
            found_skills.add(skill)
    return list(found_skills
```

Function Workflow:

1. Accept resume text and skill list as inputs
2. Convert text to lowercase for case-insensitive matching
3. Initialize empty set to store found skills (avoids duplicates)
4. Iterate through each skill in the skill list
5. Use regular expressions with word boundaries to match whole words only
6. Add matched skills to the set
7. Return list of unique skills found

Key Technical Decisions:

- **Word boundaries (\b):** Ensures "Python" matches "Python" but not "Pythonic"
- **re.escape():** Handles special characters in skill names (e.g., "C++")
- **Set data structure:** Automatically prevents duplicate skill entries
- **Lowercase conversion:** Handles case variations (Python, python, PYTHON)

---

6. Testing and Results

6.1 Testing Methodology

The parser was tested on 10 random resumes from the dataset. For each resume:

1. automatically extracted talents from the parser
2. Actual skills were found by a manual review; the results were compared to determine correctness.

6.2 Test Results

Resume_ID	Category	Extracted	Actual	Correct	Accuracy
32	HR	6	12	6	50
290	INFORMATION-TECHNOLOGY	10	11	10	90.9
875	FITNESS	5	7	5	71.4
859	FITNESS	2	5	2	40
637	BUSINESS-DEVELOPMENT	6	6	6	100
1052	SALES	1	6	1	16.7
1223	CONSULTANT	2	6	2	33.33
2299	ARTS	2	6	2	33.33
2082	PUBLIC-RELATIONS	4	7	4	57.4
1461	CHEF	8	11	8	72.7

Summary Statistics:

- Total resumes tested: 10
- Average skills per resume:
- Minimum skills found: 1
- Maximum skills found: 10
- **Average Parser Accuracy: 56.6%**

6.3 Performance Analysis

Strengths Observed:

- Parser was able to recognize common technical talents.
- Excellent precision on properly structured resumes
- Quick processing (less than one second per resume)

Limitations Identified:

- A low average parser accuracy of 56.6% was found.
- Might overlook skill variations that aren't listed in the keywords
- Lacks context awareness (for example, "no Python experience" could be equivalent to "Python").

6.4 Accuracy Calculation Method

Accuracy was calculated using the formula:

Accuracy = (Correctly Identified Skills / Total Actual Skills) × 100%

For each resume, skills correctly identified by the parser were compared against manually verified skills present in the resume text.

---

7. Conclusion

7.1 Phase 1 Summary

Phase 1 has been successfully completed with all objectives achieved:

✓ **Data collection:** Three extensive datasets including 2484 resumes, 123849 job advertisements, and 213768 job-skill mappings were acquired from Kaggle.

✓ **Data Exploration:** Extensive analysis is done to comprehend distributions, data structures, and quality concerns.

✓ **Data Cleaning:** A methodical cleaning procedure was used, and the outcome was 35 skill mappings with zero null values, 2484 clean resumes, and 122124 clean job posts.

✓ **Parser Development:** A functional resume parser with 117 comprehensive skills was developed utilizing a keyword matching strategy.

✓ **Testing:** Parser tested on 10 random resumes achieving 56.6% average accuracy.

7.2 Key Achievements

- Created pristine, well-organized datasets that are prepared for machine learning
- Created a system for extracting working skills that meets accuracy goals.
- Produced documentation and reusable code.
- Determined what needs to be improved in Phase 2.

7.3 Future Work

- Use sophisticated NLP methods (transformers, spaCy NER, etc.).

- Make a training dataset with job-resume matches labeled.
- Create and train a variety of machine learning models, such as neural networks, random forests, and logistic regression.
- Evaluate model performance and choose the most effective strategy.
- Create a method for ranking recommendations.

7.4 Deliverables Completed

All Phase 1 deliverables have been completed and are available in the project repository:

- Cleaned datasets (CSV files)
- Data exploration notebook
- Resume parser implementation
- Test results and accuracy analysis
- Complete documentation

---

8. References

1. Resume Dataset - Kaggle_ResumeData_Link
2. LinkedIn Job Postings Dataset - Kaggle_JobPosting_Link
3. Python Documentation: https://docs.python.org
4. Pandas Documentation: https://pandas.pydata.org
5. Project GitHub Repository: Repository Link
