Standardization of Jobs and Skill

with RChilli

June 2024 | Version 1.0

Purpose Statement

This document details integrating **Standardization of Jobs and Skill** use case leveraging RChilli Taxonomy.

Disclaimer:

Contents

Introduction

The **Standardization of Jobs and Skills** feature aims to ensure uniformity and standardization of job titles and skills across all candidates in the database.

By utilizing the RChilli Taxonomy API, job titles and skills are enriched and normalized with synonyms, ontology, and relationships between skills and job titles. This standardization enhances search capabilities, allowing recruiters to find the most relevant candidates more efficiently.

The **Standardization of Job Profile and Skill** feature works by:

1. **Uploading Candidate Information:** Candidate resumes are uploaded to the system.
2. **Parsing and Normalizing Data:** The system processes the uploaded resumes using the Resume Parser API, standardizing job profiles and skills using Taxonomy API.
3. **Enriching Data:** The standardized data is enriched with synonyms, ontology, and skills-to-job titles relationships, making the information more comprehensive and searchable.

Impact: Enhanced overall search results by up to 90%.

Key Components and Flow

1. **Resume Upload**:
    a. **Action**: Candidates submit their resumes to the system.
    b. **Response**: The resumes are stored and queued for processing.
2. **Data Parsing and Normalization**:
    a. **Action**: The system parses the uploaded resumes using the RChilli Resume Parser API and standardizes job profiles and skills using Taxonomy API.
    b. **Response**: The job profiles and skills are standardized and stored in the database.
3. **Data Enrichment**:
    a. **Action**: The system enriches the standardized data with synonyms, ontology, and skills-to-job titles relationships.
    b. **Response**: The enriched data is stored and made available for search.
4. **Search Enhancement**:
    a. **Action**: Recruiters perform candidate searches.
    b. **Response**: The system provides enhanced search results based on standardized and enriched data.

Benefits

1. **Improved Search Accuracy**: Standardizes job profiles and skills, improving the accuracy of search results by up to 90%.
2. **Efficiency**: Saves time for recruiters by simplifying the search process.
3. **Consistency**: Ensures uniformity in job titles and skills across all candidate profiles.
4. **Comprehensive Data**: Enriches data with synonyms, ontology, and skills-to-job titles relationships, making it more comprehensive and searchable.
5. **Enhanced Data Quality**: Enriches data with additional context and relationships, improving overall data quality.

User Stories

1. **Upload Resume**:
    a. **As a candidate**, I want to upload my resume so that my skills and job profile can be standardized and made searchable.
2. **Parse and Standardize Data**:
    a. **As a system**, I need to parse the resume and standardize the job profile and skills to improve search capabilities.
3. **Enrich Data**:
    a. **As a system**, I need to enrich the standardized data with synonyms, ontology, and skills-to-job titles relationships to make it more comprehensive.
4. **Enhanced Search**:
    a. **As a recruiter**, I want to search for candidates with standardized job profiles and skills to get more accurate and relevant search results.

Example Scenario

Scenario 1: -

1. **Scenario:** A recruiter is looking for candidates with "Software Engineer" experience.
2. **Action:** The recruiter searches for "Software Engineer".
3. **Response:** The system uses standardized job titles and enriched skills data to return relevant candidates.
4. **Outcome:** The recruiter receives a list of candidates with varying titles like "Software Developer" and "Software Programmer", all normalized under "Software Engineer".

Scenario 2: -

1. **Scenario**: A recruiter named Jane is looking for candidates with specific skills and job profiles.
2. **Action**: Jane uploads resume to the system, which standardizes and enriches the data.

3. **Response**: Jane performs a search and receives accurate and relevant results based on the standardized and enriched data.
4. **Outcome**: Jane quickly finds the candidates she needs, improving her efficiency and productivity.

User Acceptance Testing (UAT)

The **Standardization of Jobs and Skills** UAT plan ensures the feature meets user requirements and business goals. It involves validating that the system meets business requirements and functions correctly from the end-user's perspective.

This phase ensures that job titles and skills are standardized effectively, and the enhanced search capabilities are performing as expected.

UAT Scenario 1: Resume Upload and Parsing

- **Objective**: Verify that the system accurately parses and standardizes the uploaded resume.
- **Preconditions**: Candidate has a resume file ready for upload.
- **Steps**:
    1. Navigate to the resume upload page.
    2. Upload a resume file.
- **Expected Result**: The resume is parsed accurately, and the job profile and skills are standardized.
- **Acceptance Criteria**:
    - Resume parsing completes without errors.
    - Standardized data matches the resume content.

UAT Scenario 2: Data Enrichment

- **Objective**: Verify that the system enriches the standardized data with synonyms, ontology, and skills-to-job titles relationships.
- **Preconditions**: Standardized data is available in the database.
- **Steps**:
    1. Trigger the data enrichment process using Taxonomy API.
    2. Check the enriched data.
- **Expected Result**: The data is enriched correctly and comprehensively.

- **Acceptance Criteria**:
    - Enriched data includes synonyms, ontology, and skills-to-job titles relationships.
    - Enriched data is accurate and comprehensive.

UAT Scenario 3: Enhanced Search Results

- **Objective**: Verify that the system provides enhanced search results based on standardized and enriched data.
- **Preconditions**: Standardized and enriched data is available in the database.
- **Steps**:
    1. Perform a candidate search.
    2. Review the search results.
- **Expected Result**: The search results are accurate and relevant.
- **Acceptance Criteria**:
    - Search results are based on standardized and enriched data.
    - Search results are accurate and relevant.

UAT Scenario 4: Error Handling for Unsupported File Formats

- **Objective**: Verify that the system handles unsupported resume file formats correctly.
- **Preconditions**: Candidate has a resume file in an unsupported format.
- **Steps**:
    1. Attempt to upload a resume file in an unsupported format.
- **Expected Result**: The system displays an appropriate error message.
- **Acceptance Criteria**:
    - The system prevents unsupported file formats from being uploaded.
    - An informative error message is displayed.

UAT Scenario 5: Performance testing for large data sets.

- **Objective:** Ensure the response time remains constant
- **Preconditions:** Database with many candidate profiles.

- **Steps:**
    a. Upload a bulk of resumes.
    b. Run the standardization process.
    c. Measure the time taken to process and search.
- **Expected Result:** The process completes within an acceptable time frame without errors.
- **Acceptance Criteria:** The time of response is less than 500s.

Test Cases

Test Case 1: Resume Upload Test

- **Objective**: Verify that a resume can be uploaded successfully.
- **Preconditions**: Candidate has a resume file ready.
- **Steps**:
    1. Navigate to the resume upload page.
    2. Click the upload button.
    3. Select and upload a resume file.
- **Expected Result**: Resume upload completes without errors.
- **Test Data**: Various resume formats (PDF, DOCX).
- **Pass Criteria**: Resume file is uploaded and available for processing.

Test Case 2: Resume Parsing and Standardization Test

- **Objective**: Verify that the uploaded resume is parsed and standardized accurately.
- **Preconditions**: Resume file has been uploaded.
- **Steps**:
    1. Upload a resume file.
    2. Check the parsed and standardized data.
- **Expected Result**: Parsed and standardized data matches the resume content.
- **Test Data**: Sample resumes with different formats and content.
- **Pass Criteria**: Parsed and standardized data is accurate and complete.

Test Case 3: Data Enrichment Test

- **Objective**: Verify that the system enriches the standardized data correctly.
- **Preconditions**: Standardized data is available in the database.
- **Steps**:
    1. Trigger the data enrichment process using Taxonomy API.
    2. Check the enriched data.

- **Expected Result**: Enriched data includes synonyms, ontology, and skills-to-job titles relationships.
- **Test Data**: Various standardized job profiles and skills.
- **Pass Criteria**: Enriched data is accurate and comprehensive.

Test Case 4: Enhanced Search Results Test

- **Objective**: Verify that the system provides accurate and relevant search results.
- **Preconditions**: Standardized and enriched data is available in the database.
- **Steps**:
    1. Perform a candidate search.
    2. Review the search results.
- **Expected Result**: Search results are accurate and relevant.
- **Test Data**: Various search queries.
- **Pass Criteria**: Search results are based on standardized and enriched data and are accurate and relevant.

Test Case 5: Error Handling for Unsupported File Formats

- **Objective**: Verify that the system handles unsupported resume file formats correctly.
- **Preconditions**: Candidate has a resume file in an unsupported format.
- **Steps**:
    1. Attempt to upload a resume file in an unsupported format.
- **Expected Result**: The system displays an appropriate error message.
- **Test Data**: Unsupported file formats (e.g., TXT, RTF).
- **Pass Criteria**: The system prevents unsupported file formats from being uploaded and displays an informative error message.

Test Case 6: Partial Match Standardization Test

- **Objective:** Ensure that partial matches of job titles and skills are also standardized.
- **Preconditions:** Candidate database with job titles and skills having partial matches.

- **Steps:**
    1. Upload resumes with partial job titles and skill names.
    2. Run the standardization process.
    3. Search for a common job title or skill.
- **Expected Result:** Partial matches are standardized under a common term.
- **Test Data:** Resumes with job titles like "Software Eng", "Dev", and skill names like "Jav", "JS".
- **Pass Criteria:** Partial job titles and skills are searchable under a common term.

Test Case 7: Synonym Standardization Test

- **Objective:** Verify that synonyms for job titles and skills are correctly standardized.
- **Preconditions:** Candidate database with job titles and skills having synonymous terms.
- **Steps:**
    1. Upload resumes with synonymous job titles and skill names.
    2. Run the standardization process.
    3. Search for a standardized term.
- **Expected Result:** Synonyms are standardized and searchable under a common term.
- **Test Data:** Resumes with job titles like "Backend Developer", "Server-side Developer" and skill names like "JavaScript", "JS".
- **Pass Criteria:** Synonyms are searchable under a common term.

Test Case 8: Multi-language Standardization Test

- **Objective:** Ensure job titles and skills in multiple languages are standardized.
- **Preconditions:** Candidate database with job titles and skills in different languages.
- **Steps:**
    1. Upload resumes with job titles and skills in various languages.
    2. Run the standardization process.
    3. Search for a standardized term in one language.
- **Expected Result:** Job titles and skills in multiple languages are standardized.
- **Test Data:** Resumes with job titles and skills in English, Spanish, French, etc.

- **Pass Criteria:** Job titles and skills are searchable under a common term regardless of the language.

Test Case 9: Real-time Data Standardization Test

- **Objective:** Verify real-time standardization of job titles and skills.
- **Preconditions:** System integrated with real-time data input.
- **Steps:**
    1. Upload resumes in real-time.
    2. Verify immediate standardization.
    3. Search for standardized terms.
- **Expected Result:** Real-time data is standardized immediately.
- **Test Data:** Real-time resume uploads.
- **Pass Criteria:** Real-time data is searchable under standardized terms immediately after upload.

Test Case 10: Validation of Standardized Data Integrity Test

- **Objective:** Ensure the integrity of standardized data.
- **Preconditions:** Standardized candidate database.
- **Steps:**
    1. Perform a series of searches and updates.
    2. Verify the integrity of standardized data post-operations.
- **Expected Result:** Standardized data remains intact and accurate.
- **Test Data:** Various search and update operations on standardized data.
- **Pass Criteria:** No integrity issues in standardized data after operations.

Note: - Please create the other error code test cases according to the Resume Parser and Taxonomy Error Codes. KC link - https://docs.rchilli.com/kc/c_RChilli_taxonomy_error_code

Miscellaneous

The integration depends on your application workflow. There might be different steps involved in your integration which may vary from application to application.

This document indicates basic steps which are involved, you can review your application and add other UAT, and test cases based upon your needs.

- **Dependencies:** Requires integration with the Taxonomy API.
- **Data Quality:** Regularly update and maintain the Taxonomy API for best results.

Integration Estimation

Disclaimer: This estimation is based on a sample application with simple workflow. This may vary depending on the complexity of your application and the skill set/experience of the team involved in the integration.

1. **Development (25hr – 30hr)**
    a. Resume Upload: 2hr
    b. Resume Parsing and Standardization using RChilli Parser API: 4hr
    c. Data Enrichment: 6-8hr
    d. Enhanced Search Algorithm: 8-10hr
    e. Recommendation Display: 3-4hr
    f. Unit Test Cases: 2-3hr
    g. Bug Fixing and Optimization: 2-3hr
2. **QA – (7hr -10hr)**
    a. UAT: 2-3hr
    b. Executing Test Cases: 5-7hr

Appendix

Taxonomy API Details:

https://docs.rchilli.com/kc/c_RChilli_taxonomy

Get API key and Endpoints:

https://docs.rchilli.com/kc/c_RChilli_taxonomy_API_Endpoints

Postman Integration and Sample Code:

https://documenter.getpostman.com/view/6003669/Szmh1GQX?version=latest#bdb6addf-ed0e-41b4-a088-dea0d9e687a7

Taxonomy API Response Schema:

https://docs.rchilli.com/kc/c_RChilli_taxonomy_Response_schema

Resume Parser Error Code:

https://docs.rchilli.com/kc/c_RChilli_taxonomy_error_code

Language Supported:

https://www.rchilli.com/languages