AI-Powered Resume Analyzer

Document Owner - Md Mahabub Alam

created on - June 10, 2025

Problem Statement

Develop an AI-powered Resume Analyzer tool to assist users in enhancing their resumes by analyzing uploaded documents and providing detailed feedback on content, structure, completeness, and sentiment. The tool outputs a structured analysis in JSON format and generates a PDF report for easy sharing.

Requirements

Input

- A resume file in PDF or plain text or Microsft Doc format

Processing Tasks

The system must:

1. Extract raw text from the uploaded resume (PDF or text or Docx).
2. Analyze the text using a Large Language Model (LLM) to:
    - Identify and categorize key sections (e.g., Summary, Skills, Experience, Education).
    - Detect missing or underdeveloped sections (e.g., no summary, sparse skills).
    - Evaluate clarity, professionalism, and completeness.
    - Assess the sentiment of the skills section (e.g., confident, neutral, vague).
3. Compute a **Resume Quality Score** (0–100) based on:
    - Section completeness (presence of key sections).
    - Content richness (use of specific, quantifiable details).
    - Clarity and professionalism (readability, tone).
    - Overall resume strengh for the role and years of experience
4. Highlight strengths (e.g., well-written sections or standout achievements).
5. Generate actionable improvement suggestions (e.g., “Add a Certifications section,” “Include metrics in Experience”).

Output

A structured JSON result containing:

```
{
  "sections_detected": ["Summary", "Skills", "Experience", "Education"],
  "missing_sections": ["Certifications", "Projects"],
  "well_written_sections": [
    "Expereince section has depth in skills",
    "Solid educational background from top tier institute"
  ],
  "resume_quality_score": 78,
  "skills_sentiment_summary": "Confident and specific, but lacks technical
     keywords",
  "improvement_suggestions": [
    "Add a Certifications section to showcase credentials.",
    "Incorporate quantifiable achievements in the Experience section."
  ]
}
```

The output must also be exportable as a PDF report summarizing the analysis for easy sharing.

Challenge Points

LLM Integration Leverage an LLM (e.g., openai, google gemini or grok or any other) to semantically analyze and evaluate resume content.

PDF/Text/Docx Extraction Accurately extract text from the supported files, handling varied layouts.

Information Extraction Parse resume structure dynamically, without relying on fixed templates.

Score Computation Develop a scoring model based on section presence, content depth and clarity.

Explainability Provide transparent reasons for score deductions and tailored suggestions.

Bonus Challenges

1. Accurately parse resume content and convert it into a standardized, code-readable JSON format, e.g.:

```
{
  "personal_info": {
    "name": "John Doe",
    "email": "john.doe@example.com"
  },
  "experience": [
    {
      "title": "Software Engineer",
      "company": "Tech Corp",
      "duration": "2020-2023",
      "details": ["Led a team of 5", "Developed 10+ features"]
    }
  ]
}
```

2. Identify and flag excessive jargon or filler phrases (e.g., “dynamic synergy”).
3. Suggest ATS-friendly formatting (e.g., use of keywords, simple headings).

Timeline and Guidelines

- **Timeline**: The task must be completed within **2 days** from the start date i.e. now
- **Guidelines**: Chose your preferred programming environment to build the solution
- **Guidelines**: Candidates should avoid using internet-based solutions or AI tools (e.g., code from online repositories or AI-generated solutions) to ensure originality. Solutions must be developed independently.

Deliverables over mail

- A functional code logic in compressed file to process and analyze resumes
- JSON output with detailed analysis and sentiment summary.
- A summary of findings over text or pdf