IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI RESUME ANALYZER

[1]Aarti, [2]Anusha Choubey, [3]Amisha Sahu, [4]Arti Verma

[1,2,3,4]*UG Student*

Department of Computer Science and Engineering

SHRI SHANKARACHARYA TECHNICAL CAMPUS

JUNWANI, BHILAI – 490 020, DISTT.: DURG

CHATTISGARH, INDIA

Abstract: In today's fast-paced world, companies receive thousands of resumes for job openings. Manually reading and evaluating each resume takes a lot of time and effort. To solve this problem, we created an **AI Resume Analyzer** that uses Artificial Intelligence to read, understand, and analyses resumes automatically. This system helps recruiters save time and choose the most suitable candidates based on the job requirements. We built this system using Python and several useful libraries. The main idea is to take resumes (usually in PDF format), extract all the text, and analyse the content using AI models. Our application is user-friendly and runs on a web interface using **Streamlit** (streamlit library), allowing recruiters to upload resumes directly from their browser.

To start, we used the **pdf2image** library to convert PDF resumes into images. This is useful because sometimes the text in PDFs is not directly selectable or searchable. By converting the pages to images, we can apply Optical Character Recognition (OCR) using **pytesseract**. This library reads the text from images, helping us extract words even from scanned documents.

We also used **pdfplumber**, a library that helps extract text, tables, and layout from PDF documents more accurately. This library is very useful when the resume is in a structured digital format. By combining both pdf2image and pdfplumber, our system can work well with all kinds of resumes — scanned or typed.

Once the text is extracted, we apply Natural Language Processing (NLP) and AI to understand the content. We use **Google Generative AI (google.generativeai)** to analyse the resume. This model can understand human language and helps evaluate important details in a resume like education, skills, experience, and achievements. For example, it can check if the candidate knows Python, has worked with machine learning, or has project experience relevant to the job.

We also use **doting** to manage secret API keys and environment variables safely. This way, sensitive information like API keys is not exposed in the code, keeping the application secure.

The steps in our AI Resume Analyzer are:

1. The user uploads a resume PDF using the web interface built with Streamlit.
2. The system checks if the PDF is scanned or typed.
3. It extracts the text using pdfplumber or pytesseract (after converting it to image with pdf2image).
4. The extracted text is passed to a Generative AI model from Google to analyse and summarize the content.
5. The results are displayed on the screen: highlighting the candidate's strengths, skills, experience, and match percentage with a job description (if given).

We also use **PIL (Python Imaging Library)** to handle images during the conversion and display process.

This project is highly helpful for HR departments, recruitment agencies, and hiring platforms. It reduces manual work, gives quick results, and provides a smart way to shortlist candidates.

In conclusion, the **AI Resume Analyzer** combines document processing and AI-powered text understanding to make hiring more efficient. With the help of modern libraries like streamlit, pytesseract, pdf2image, pdfplumber, and google.generativeai, we created a tool that reads resumes like a human and makes recruitment smarter and faster.

***Key Words*:** NLP (Natural Language Processing) Resume Parsing, Resume Analysis, Google Gemini AI, Streamlit Application, Skill Gap Detection, Career Recommendation System, OCR Resume Parsing

I. INTRODUCTION

In the digital age, where the first impression often begins with a resume, students face the critical task of crafting documents that effectively showcase their potential. A resume is not just a summary of academic achievements and experiences—it is a strategic tool that communicates one's skills, competencies, and suitability for a particular role. For students who are new to professional writing, creating an impactful and well-organized resume can be challenging. Mistakes in formatting, lack of clarity, missing keywords, or irrelevant content can significantly reduce their chances of being shortlisted for internships, jobs, or higher studies.

To bridge this gap, the Resume Analyzer has been developed as a smart assistant tailored for students. This system helps them evaluate their resumes against standard benchmarks and industry expectations. It identifies strengths, weaknesses, and missing components through automated analysis, offering detailed and constructive feedback. The tool uses advanced techniques such as Natural Language Processing (NLP) to understand and process resume content, and machine learning algorithms to match resumes against relevant job descriptions or academic profiles.

The primary goal of this project is to provide students with a user-friendly platform that not only analyses but also educates them about resume best practices. By integrating real-time feedback, scoring mechanisms, and improvement suggestions, the Resume Analyzer acts as a virtual career guide. This initiative contributes to better preparedness among students and enhances their confidence during recruitment or application processes.

II. LITERATURE REVIEW

Several studies and tools have explored the use of automation in resume evaluation. Traditional systems rely on keyword matching or template-based scoring, but modern resume analyzers incorporate Natural Language Processing (NLP) and Machine Learning (ML) for contextual analysis. Existing platforms like Rezi, Zesty, and Resume Worded offer scoring based on industry trends, but they are often premium services and not tailored to student resumes.

For this project, various libraries and tools were considered:

Python-docx / PyPDF2 / pdfminer: For extracting content from .docx and .pdf files.

spaCy / NLTK: Used for tokenization, part-of-speech tagging, and named entity recognition.

scikit-learn: Applied for keyword extraction and similarity matching between resumes and job descriptions.

Tkinter / Flask: Used for creating a GUI or web interface for the user.

matplotlib / seaborn: Used for visualizing analytics such as skill distribution or section completeness.

These tools together helped design a system that is fast, scalable, and student friendly.

III. Problem Statement

Many students, particularly freshers, face difficulties in crafting resumes that meet professional standards. Due to a lack of proper guidance, they often submit resumes with poor structure, irrelevant content, or weak descriptions of skills and experiences. Manual review processes are time-consuming and inconsistent. Therefore, there is a need for an automated system that can assist students in evaluating and enhancing their resumes through intelligent analysis and suggestions.

IV. Advantages

1. Timesaving: Students can instantly get feedback instead of waiting for mentors or placement coordinators.
2. Personalized Suggestions: Tailored feedback helps students identify what to add or remove based on their goals.
3. Objective Scoring: Removes human bias by using consistent evaluation metrics.
4. Skill Gap Identification: Highlights missing skills in comparison to targeted job profiles.
5. Learning Tool: Educates users about resume-building techniques and common industry practices.

V. Discussion

The Resume Analyzer provides a practical solution to a common problem among students. During implementation, challenges included accurately parsing resumes of varying formats and detecting inconsistencies in content structure. NLP techniques allowed the system to intelligently categorize sections like education, experience, and skills.

One notable enhancement was the integration of a job description (JD) comparison module. This allowed the system to calculate how well a resume aligns with a given JD using cosine similarity and TF-IDF vectors. Results showed that resumes with relevant keywords scored higher, emphasizing the need for keyword optimization.

Through testing with real student resumes, it was evident that many users were unaware of missing elements like project details, action verbs, or certifications. The Resume Analyzer successfully addressed these issues and demonstrated the potential for integration into university placement portals or resume workshops.

VI. Acknowledgement

We would like to express our sincere gratitude to our faculty mentor, [Mentor's Name], for their continuous guidance and support throughout the development of this project. We are also thankful to our department and fellow students who contributed valuable feedback during testing. Special thanks to open-source contributors and developers whose libraries and frameworks made this project possible.

VII. Conclusion

The Resume Analyzer is a smart, scalable, and effective tool that empowers students to improve their resumes with confidence. By providing personalized feedback and aligning resumes with industry expectations, it prepares students for success in the job market. Future enhancements may include integration with LinkedIn profiles, support for multiple languages, and real-time resume building assistance through AI chatbots.

VIII. REFERENCES

1. Streamlit Documentation
   Streamlit, Retrieved from https://docs.streamlit.io. This platform was used to build an interactive web application interface,
   allowing for rapid deployment and user-friendly experience.

2. Google Gemini AI
   Google Generative AI, Retrieved from https://cloud.google.com/generative-ai. The AI model used for generating content
   and analyzing resumes.

3. pdfplumber
   pdfplumber GitHub Repository, Retrieved from https://github.com/jsvine/pdfplumber. A Python library that helped extract
   text from PDF files efficiently.

4. pytesseract
   pytesseract Documentation, Retrieved from https://pypi.org/project/pytesseract/. A Python wrapper for Google's Tesseract-
   OCR Engine, used for extracting text from images in resumes.

5. Coursera: Python for Everybody
   Coursera Course, Retrieved from https://www.coursera.org/specializations/python. A recommended online course for
   enhancing Python programming skills.