Intelligent Resume Screening System using AI

P Devamma[1], D.Murali[2]
[1]PG Student, QUBA College of engineering and technology
[2]Associate Professor, QUBA College of engineering and technology

Abstract: The AI Resume Screener is an innovative, AI-powered application developed to revolutionize the traditional recruitment process by automating resume screening. In today's fast-paced hiring environment, HR professionals and recruiters often struggle with manually reviewing large volumes of resumes, which is both time-consuming and prone to human error. To address this issue, the AI Resume Screener utilizes Natural Language Processing (NLP) and Machine Learning (ML) techniques to analyse resumes, match them with job descriptions, and rank candidates based on compatibility, thereby streamlining the hiring process and improving overall efficiency. The system is implemented using Streamlit, a Python-based framework that allows the creation of lightweight and interactive web applications. It features a simple, intuitive dashboard where users can directly upload resumes in various formats including PDF, DOCX, and TXT. Once uploaded, the resumes are parsed using tools like PyPDF2, pdfminer, and docx2txt to extract key details such as the candidate's name, contact information, skills, education, work experience, and certifications. These extracted elements are then processed using SpaCy and NLTK, which enable advanced text analysis and ensure accurate identification of relevant data. A core functionality of the system is the AI- driven candidate ranking.

***Index Terms*— Artificial Intelligence (AI), Natural Language Processing (NLP), Machine Learning (ML), Resume Screening, Candidate Ranking, Job Matching, Automated Recruitment System, Text Mining, Information Extraction, Document Parsing.**

1. INTRODUCTION

1.1 Overview

Recruitment is a fundamental yet challenging process for any organization. In a competitive job market, companies often receive hundreds or even thousands of resumes for a single job posting. Manually reviewing these resumes is not only labor-intensive but also prone to human error, unconscious bias, and inconsistencies. The traditional hiring process struggles to maintain both speed and accuracy, often resulting in missed opportunities and delayed hiring decisions. To address these limitations, the AI Resume Screener project introduces an automated system that leverages Natural Language Processing (NLP) and Machine Learning (ML) to efficiently and fairly screen candidate resumes. The system is designed to extract key information from resumes, evaluate candidate profiles in relation to specific job descriptions, and rank applicants based on relevance. The project is implemented using Streamlit, a modern and lightweight Python framework that allows for the creation of an interactive dashboard for resume analysis. It supports resume parsing, keyword and skill matching, candidate ranking, customizable filters, and even Applicant Tracking System (ATS) optimization. This solution empowers HR professionals to make faster, more accurate, and bias-free hiring decisions, ultimately streamlining the entire recruitment pipeline.

1.2 History

While a dedicated "History" section isn't explicitly present, the documents implicitly address the history of resume screening by contrasting traditional methodologies with the proposed AI- driven solution. Traditional resume screening is characterized as a time-intensive and laborious process, often requiring HR professionals to manually sift through numerous applications, which can be a source of inefficiency and potential for human error and bias. Furthermore, traditional Applicant Tracking Systems (ATS) and keyword-based filters, while intended to aid in this process, still present challenges such as inaccurate keyword matching and a lack of contextual understanding. The AI Resume Screener is thus positioned as a necessary evolution in resume screening, designed to overcome these historical limitations and modernize the recruitment workflow.

1.3 Problem Statement The AI Resume Screener directly addresses a core set of problems inherent in

traditional resume screening practices. These problems include the time-consuming nature of manually reviewing large volumes of resumes, the repetitive and tedious nature of the task, and the potential for human bias and errors to influence candidate selection. Moreover, traditional systems often struggle with effectively matching resumes to job descriptions, leading to poor hiring decisions, and they typically lack the ability to personalize screening processes to specific job roles or industries. Even with the adoption of Applicant Tracking Systems (ATS), challenges persist, such as inaccurate keyword matching, lack of contextual understanding of candidate skills and experience, and inconsistent shortlisting criteria. In essence, the central problem is the inefficiency, inaccuracy, and potential unfairness of traditional resume screening, which necessitates an AI-driven solution. Despite offering partial automation, existing resume screening systems face several significant challenges:

Time-Consuming: HR professionals spend a considerable amount of time reading and analyzing resumes, especially when dealing with large applicant pools. This often delays hiring timelines and increases administrative workload.

Lack of Contextual Understanding: Keyword-based systems are unable to understand the context of words. For example, a candidate may list "project leadership" but be overlooked if the system searches only for "team management."

1.4 Research Motivation

The development of the AI Resume Screener is motivated by the need to resolve the significant shortcomings of traditional resume screening methods and to modernize the recruitment process. The primary motivations include increasing efficiency by automating time-consuming manual tasks, improving accuracy in candidate selection to identify the most qualified individuals, and reducing bias to promote fairer hiring practices. Furthermore, there's a strong drive to provide a more objective and data-driven approach to resume screening, leveraging AI to analyze candidate qualifications in a consistent and insightful manner. Ultimately, the research is motivated by the desire to create a tool that empowers HR professionals and recruiters to make better, faster, and more equitable hiring decisions.

1.5 Applications

The AI Resume Screener is designed to be a versatile tool with broad applicability across various sectors and organizations involved in the hiring process. Its core functionality of automating and enhancing resume screening makes it highly beneficial for HR departments within companies of all sizes, as it can streamline their internal recruitment procedures and improve efficiency. Recruitment agencies can also leverage the AI Resume Screener to expedite candidate sourcing and selection for their clients, enabling them to provide faster and more effective services. Furthermore, online hiring platforms can integrate the AI Resume Screener to enhance their offerings, providing a more sophisticated and user-friendly experience for both job seekers and employers. In essence, any entity involved in the recruitment process stands to gain from the AI Resume Screener's ability to automate, accelerate, and optimize the identification and selection of qualified candidates.

II LITERATURE SURVEY

Introduction and the Evolution of Resume Screening

Resume screening is the foundational process within recruitment, involving the evaluation of candidate resumes to ascertain their initial suitability for a given job role. Its importance lies in its role as the gatekeeper, determining which candidates advance to subsequent stages of the hiring process. As the volume of job applications continues to surge, organizations face increasing pressure to adopt efficient and scalable screening methodologies.

Core AI Technologies and Techniques

Natural Language Processing (NLP) plays a central role in AI-based resume screening systems, providing the ability for computers to understand and process human language. NLP techniques are essential for analyzing the textual content of resumes, enabling systems to go beyond simple keyword matching. Text extraction is a fundamental NLP task, involving the extraction of raw text from resumes in various formats such as PDF, DOCX, and TXT. Libraries and tools like PyPDF2, pdfminer, and docx2txt are commonly used for this purpose.

III EXISTING METHODOLOGY

3.1 Existing System

The existing system for resume screening predominantly relies on traditional methods, often involving manual review by HR professionals and recruiters. Even with the adoption of Applicant Tracking Systems (ATS), these methods present several challenges. A key issue is the time- consuming nature of manually screening numerous resumes, which slows down the hiring process. Many traditional systems depend heavily on keyword matching to filter candidates. This approach can be inaccurate, as it may reject qualified individuals who use different terminology or resume formats. Furthermore, manual screening introduces the risk of human bias, potentially leading to unfair candidate selection.

This manual effort is necessary to:
Assess Candidate Qualifications: Determine if candidates possess the required skills, experience, and education.
Evaluate Soft Skills: Try to gauge a candidate's communication, problem-solving, and other non-technical skills (though this is difficult from a resume alone).

3.2. DRAWBACKS

1. Time-Consuming Nature of Manual Review
Existing systems heavily rely on HR professionals and recruiters to manually read and assess each resume. This process becomes extremely time-consuming, especially when dealing with a large number of applications for a single job opening. The time spent on manual screening detracts from other important HR functions and slows down the overall hiring process.

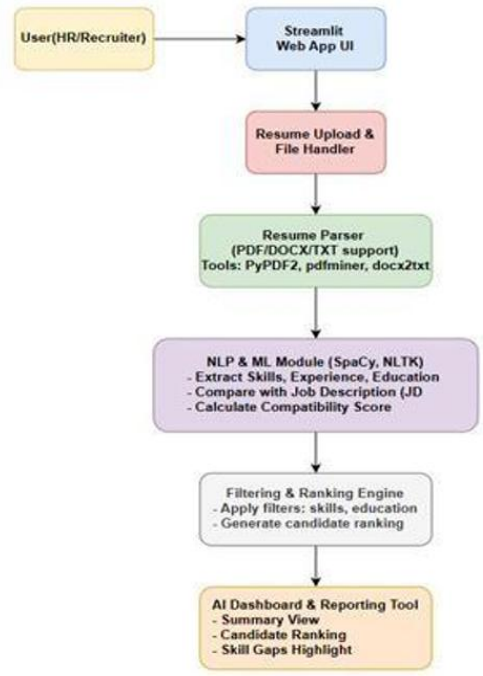
2. Inaccuracy and Limitations of Keyword Matching
Many traditional systems, including some ATS, use keyword matching as a primary method for filtering resumes. This approach involves scanning resumes for specific keywords related to the job description.

3. Introduction of Human Bias Manual resume screening is susceptible to unconscious bias on the part of recruiters. Factors such as the candidate's name, gender, ethnicity, or other demographic information can unintentionally influence the screening decision. This bias can lead to unfair hiring practices and the exclusion of potentially valuable candidates.

IV. PROPOSED METHODOLOGY

4.1 Overview

BLOCK DIAGRAM



4.1. BLOCK DIAGRAM

User Interface (Streamlit Web App UI):
The system provides a user interface, built with Streamlit, for HR personnel and recruiters to interact with.

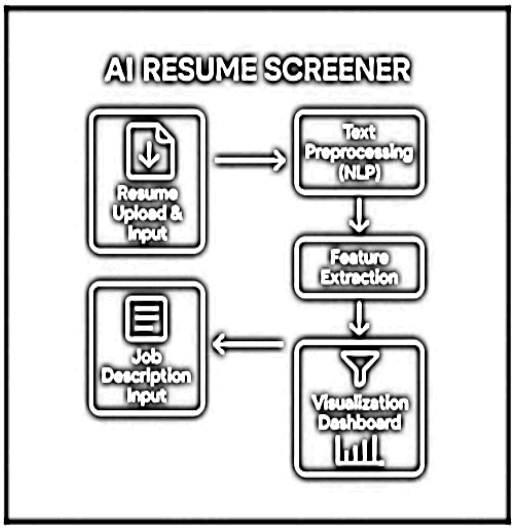
Resume Upload & File Handler:
The system allows users to upload resumes in various formats.

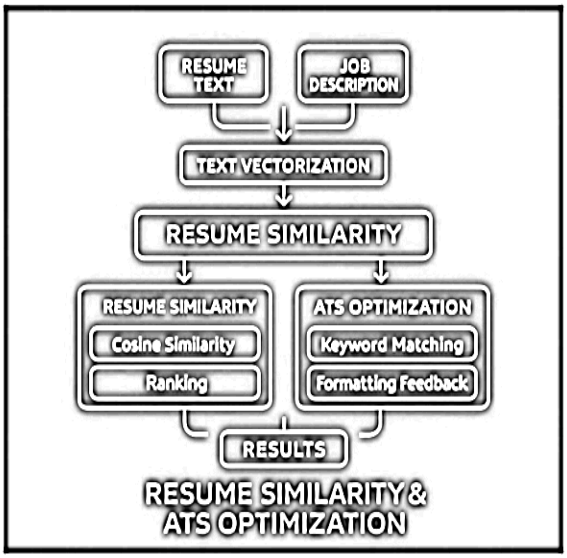
Resume Parser:
The system uses tools like PyPDF2, pdfminer, and docx2txt to parse and extract data from resumes in PDF, DOCX, and TXT formats.

4.2 Proposed system

The proposed system is an AI-powered resume screener developed using Streamlit, aimed at automating and optimizing the initial stages of recruitment. This intelligent tool utilizes Natural Language Processing (NLP) and Machine Learning

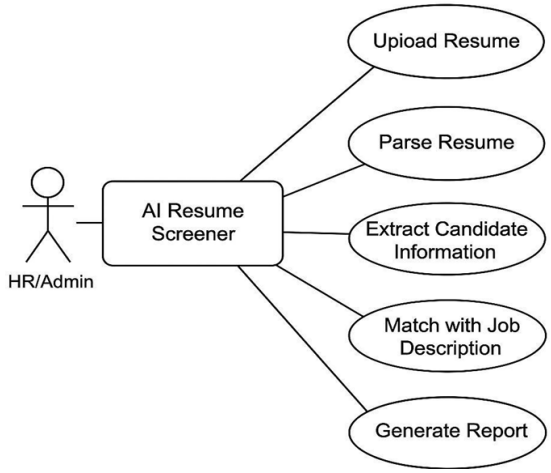
(ML) techniques to efficiently process and evaluate candidate resumes against specific job descriptions. The system is designed to reduce manual effort in resume screening by automatically parsing documents, extracting relevant information, and ranking candidates based on compatibility scores.





V. UML REPRESENTATION

Unified Modelling Language (UML) diagrams play a crucial role in the design and development of the AI Resume Screener by providing a visual representation of system architecture, behaviour, and interactions. These diagrams help in better understanding, planning, and maintaining the system.



Use Cases (Ellipses):

These ovals represent the different functionalities or tasks that the HR/Admin actor can perform using the AI Resume Screener like below

Upload Resume: Allows the HR/Admin to upload resume files into the system.

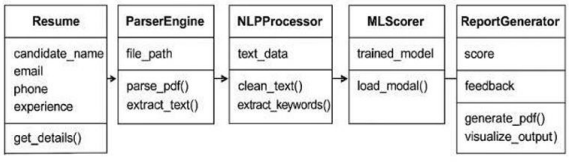
Parse Resume: The system analyzes the uploaded resume to extract relevant data.

Extract Candidate Information: The system identifies and pulls out key information from the parsed resume, such as skills, experience, education, etc.

Match with Job Description: The system compares the extracted candidate information with the details of a specific job description.

Generate Report: The system creates a report, likely summarizing the matching results or other relevant data.

5.2 Class Diagram:



VI SOFTWARE ENVIRONMENT

This chapter provides a comprehensive and detailed overview of the software environment meticulously chosen and configured for the development and deployment of our intelligent AI-based Resume Screener project. A robust and well-documented

software environment is paramount for ensuring seamless collaboration among development team members, facilitating efficient deployment across various infrastructure setups, and guaranteeing the long-term maintainability and scalability of our system. This chapter serves as a definitive guide to the technological underpinnings of our project.

6.1 Python: The Core Programming Language

Python serves as the bedrock upon which our entire AI-based Resume Screener is built. Its selection was a strategic decision based on its unique blend of power, readability, and an exceptionally rich ecosystem tailored to the demands of modern software development, particularly in the domains of Artificial Intelligence, Natural Language Processing, and web application development.

Figure 6.5.1. Python official site

VII SYSTEM REQUIREMENTS

7.1 Software Requirements:

Components	Description
Programming Language	Python 3.11
Framework	Streamlit (for UI/dashboard)
Libraries	pandas, numpy, nltk or spaCy, scikit-learn, pyresparser, resume-parser, joblib, pdfplumber, docx2txt
ML/NLP Tools	custom ML model for resume ranking
OS Compatibility	Windows
Browser	Chrome (to view Streamlit dashboard)
IDE/Editor	VS Code / PyCharm / Jupyter Notebook

7.2 Hardware Requirements:

Requirement	Minimum	Recommended
Processor	Dual-core 1.8 GHz	i5 or higher
RAM	4 GB	8 GB or more
Storage	1 GB free space	SSD with 5 GB or more
Display	1024x768resolution	1366x768 or higher
Internet	Required to run	

VIII REQUIREMENT ANALYSIS

8.1 Functional Requirements

Output Design

- Display resume analysis results on a Streamlit dashboard.
- Show candidate ranking based on similarity score.
- Provide skill match percentage for each candidate.
- Offer visual analytics using charts (e.g., bar chart, pie chart, score graph).
- Export matched candidate results as downloadable CSV/Excel file.

Output Definition

1. Candidate Name

2. Extracted Skills

3. Experience (if mentioned)

4. Contact Details (Email, Phone)

8.2 SOURCE CODE

APP.PY:

```
import streamlit as st
import os
import pandas as pd
st.set_page_config(
page_title="AI Resume Screener",
page_icon=")
 ‘",
layout="wide",
initial_sidebar_state="expanded"
```

```
)
if 'uploaded_resumes' not in st.session_state:
st.session_state.uploaded_resumes = []
if 'parsed_resumes' not in st.session_state:
st.session_state.parsed_resumes = []
if 'resume_df' not in st.session_state:
st.session_state.resume_df = pd.DataFrame()
if 'job_description' not in st.session_state:
st.session_state.job_description = ""
if 'ranked_candidates' not in st.session_state:
st.session_state.ranked_candidates = pd.DataFrame()
st.title("AI Resume Screener")
st.markdown("""
if st.session_state.parsed_resumes:
st.success(f" ■ {len(st.session_state.parsed_resumes)
} resumes have been uploaded and
parsed")
st.subheader("Uploaded Candidates")
if not st.session_state.resume_df.empty:
 ) Don't forget to add a job description to compare
resumes against")
```

IX RESULTS AND DISCUSSION

Implementation Description

The implementation of the AI Resume Screener System is divided into multiple phases to ensure the development of a robust, scalable, and efficient application that can analyze resumes, match them with job descriptions, and rank candidates effectively. The system uses Natural Language Processing (NLP) and Machine Learning (ML) to automate resume screening and is deployed using Streamlit for real-time, interactive web access.

Environment Setup and Tool Installation

A suitable development environment plays a crucial role in the smooth execution and efficiency of a software project. In the case of the AI Resume Screener System, setting up the right tools, libraries, and platforms is essential to facilitate natural language processing (NLP), machine learning (ML), resume parsing, and a user-friendly interface via Streamlit. This section explains the environment and installation process required to develop and run the system effectively.

Fig 10.2(5). Comparision of Candidates(Bar chart)

X CONCLUSION

The AI-Based Resume Screener System has successfully demonstrated how Artificial Intelligence and Natural Language Processing (NLP) can be leveraged to enhance the recruitment process by automating the initial screening of resumes. By utilizing Python, NLP libraries (spaCy, NLTK), and resume parsing tools (PyPDF2, pdfminer.six, docx2txt), the system efficiently extracts and processes candidate information from various resume formats.

REFERENCE

[1] Pradeep Kumar Roy, Sarabjeet Singh Chowdhary and Rocky Bhatia, "A Machine Learning approach for automation of Resume Recommendation system", International Conference on Computational Intelligence and Data Science (ICCIDS 2019).

[2] Adem Golec and Esra Kahya, "A fuzzy model for competency-based employee evaluation and selection", Computers & Industrial Engineering, vol. 52, pp. 143-161, 2007.

[3] Rahul, S. Adhikari and Monika, "NLP based Machine Learning Approaches for Text Summarization", 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), pp. 535-538, 2020.

[4] G. Silva, L. Rafael Lins, S. J. Simske and M. Riss, "Automatic Text Document Summarization Based on Machine Learning", DocEng'15 Lausanne, 2015.

[5] Begum Mutlu, Ebru A. Sezer and M. Ali Akcayol, "Candidate sentence selection for extractive text summarization", Information Processing & Management, vol. 57, no. 6, pp. 102359, 2020.