

AI-powered Resume Screening and Ranking System

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning

with

TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Raghu Ebbili, raghuraghava118@gmail.com

Under the Guidance of

Abdul Aziz Md

Master Trainer, Edunet Foundation

ACKNOWLEDGEMENT

We would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this thesis work.

Firstly, we would like to thank my Mentor Abdul Aziz Md for being a great mentor and the best adviser I could ever have. His advice, encouragement and the critics are a source of innovative ideas, inspiration and causes behind the successful completion of this project. The confidence shown in me by him was the biggest source of inspiration for me. It has been a privilege working with him for the last one year. He always helped me during my project and many other aspects related to the program. His talks and lessons not only help in project work and other activities of the program but also make me a good and responsible professional.

ABSTRACT

In today's highly competitive job market, recruiters face challenges in manually screening numerous resumes for a single job posting. This process is time-consuming and prone to biases. To address this issue, we propose an AI-powered Resume Screening and Ranking System that automates the evaluation and ranking of resumes based on job descriptions.

The primary objectives of this project are to automate the resume screening process, rank candidates based on job relevance, and enhance recruitment efficiency. The system utilizes Natural Language Processing (NLP) techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) and cosine similarity to compare job descriptions with resumes. By extracting text from resumes using PyPDF2 and preprocessing it, the system converts textual data into numerical vectors. The similarity scores are then computed to rank resumes effectively.

The methodology includes extracting text from uploaded resumes, preprocessing data to remove noise, converting text into vector representations, and calculating similarity scores using cosine similarity. The ranked results are displayed in an interactive user interface built with Streamlit, allowing recruiters to easily analyze and shortlist candidates.

Key results of this system demonstrate improved accuracy in candidate selection, reduced manual effort, and an efficient, data-driven approach to resume screening. The implementation enhances the hiring process by providing recruiters with a ranked list of candidates best suited for a job role.

In conclusion, this AI-driven tool streamlines recruitment by automating resume evaluations and reducing human bias. Future enhancements may include deep learning-based NLP models, integration with Applicant Tracking Systems (ATS), and multi-format resume support to further optimize hiring workflows.

TABLE OF CONTENT

| | | |
|--|-------|----------|
| Abstract | | I |
| Chapter 1. Introduction | | 1 |
| 1.1 Problem Statement | | 1 |
| 1.2 Motivation | | 1 |
| 1.3 Objectives | | 2 |
| 1.4 Scope of the Project | | 2 |
| Chapter 2. Literature Survey | | 3 |
| Chapter 3. Proposed Methodology | | |
| Chapter 4. Implementation and Results | | |
| Chapter 5. Discussion and Conclusion | | |
| References | | |

LIST OF FIGURES

| Figure No. | Figure Caption | Page No. |
|------------|---------------------------------------|----------|
| Figure 1 | Extracting Text from Resumes | 7 |
| Figure 2 | Converting Text to TF-IDF Vectors | 8 |
| Figure 3 | Computing Cosine Similarity & Ranking | 8 |
| Figure 4 | Displaying Results in Streamlit | 9 |
| Figure 5 | Expected Output | 9 |
| Figure 6 | | |
| Figure 7 | | |
| Figure 8 | | |
| Figure 9 | | |

LIST OF TABLES

[illegible]

CHAPTER 1

Introduction

1.1 Problem Statement:

Recruiters receive a large volume of resumes for job openings, making manual screening a tedious and time-consuming process. Traditional screening methods are inefficient, subjective, and prone to human bias, often leading to delays in hiring and the possibility of missing out on qualified candidates. This project aims to address these challenges by developing an AI-powered system that automates resume evaluation and ranks candidates based on their suitability for a given job description.

1.2 Motivation:

The increasing number of job applicants and the demand for an efficient recruitment process inspired this project. Organizations need a fast and reliable way to shortlist candidates while ensuring fairness and accuracy. This AI-powered system minimizes human effort and enhances decision-making by leveraging natural language processing (NLP) and machine learning techniques. Potential applications include recruitment agencies, corporate HR departments, and online job portals, significantly impacting hiring efficiency.

1.3 Objective:

The primary objectives of this project are:

To automate the resume screening process and eliminate manual effort.

To rank resumes based on job relevance using NLP techniques.

To enhance fairness and reduce bias in candidate evaluation.

To provide recruiters with an interactive and data-driven ranking system.

1.4 Scope of the Project:

The project focuses on developing an AI-based resume screening system that processes resumes in PDF format, extracts relevant textual information, and ranks candidates based on job descriptions. Limitations include:

- The system currently supports only text-based PDF resumes.
- It relies on keyword matching and similarity scoring without deep contextual understanding.
- The model does not evaluate candidate soft skills or interview performance.
- Future expansions may include multi-format resume processing, integration with applicant tracking systems, and advanced AI models for better contextual analysis.

CHAPTER 2

Literature Survey

2.1 Review of Relevant Literature:

Several studies have explored the use of AI and NLP techniques in recruitment and resume screening. Traditional hiring methods rely on human recruiters, leading to inefficiencies and biases. Existing research highlights the effectiveness of AI-based resume screening, utilizing machine learning algorithms and text similarity techniques to improve hiring outcomes.

2.2 Existing Models and Methodologies:

Previous approaches include rule-based keyword matching, which is widely used in Applicant Tracking Systems (ATS). More advanced methods leverage machine learning models, including:

- **TF-IDF and Cosine Similarity:** These techniques quantify text similarity, helping match resumes with job descriptions.
- **Word Embeddings (Word2Vec, GloVe, BERT):** These models capture contextual meaning, improving accuracy in candidate ranking.
- **Deep Learning Models:** Neural networks such as LSTMs and Transformers enhance resume parsing and semantic matching.

2.3 Gaps and Limitations in Existing Solutions:

Despite advancements, existing systems face limitations, such as:

- 2 **Keyword Dependency:** Many ATS systems rely on keyword matching, missing contextual relevance.
- 3 **Lack of Deep Understanding:** Traditional models struggle with nuanced resume-job description relationships.
- 4 **Bias and Fairness Issues:** AI models may inherit biases from training data, affecting decision-making.

CHAPTER 3

Proposed Methodology

3.1 System Design

The system follows a structured pipeline that automates the resume screening and ranking process. The key components include:

4 Resume Upload & Text Extraction:

4.1 Users upload multiple resumes in PDF format.

4.2 PyPDF2 extracts text from the resumes.

5 Preprocessing & Feature Extraction:

5.1 NLP techniques (tokenization, stop-word removal) clean the text.

5.2 TF-IDF vectorization converts resumes and job descriptions into numerical vectors.

6 Similarity Computation & Ranking:

6.1 Cosine similarity measures the relevance of each resume to the job description.

6.2 Resumes are ranked based on their similarity scores.

7 User Interface (UI):

7.1 Built using Streamlit, the UI displays ranked resumes with scores.

7.2 Requirement Specification

To implement the AI-powered resume screening system, the following requirements must be met:

3.2.1 Hardware Requirements:

- **Processor:** Intel Core i5 or higher
- **RAM:** 8GB minimum (16GB recommended)
- **Storage:** At least 20GB of free space
- **GPU (Optional):** Recommended for future deep learning enhancements

3.2.2 Software Requirements:

- **Operating System:** Windows, macOS, or Linux
- **Programming Language:** Python 3.x
- **Libraries & Dependencies:**
 - PyPDF2 (for PDF text extraction)
 - Scikit-learn (for TF-IDF and Cosine Similarity)
 - Streamlit (for UI development)
 - Pandas (for data handling)
 - NumPy (for numerical operations)

CHAPTER 4

Implementation and Result

Implementation Details

4.1 System Architecture

The system follows a structured pipeline to automate the resume screening and ranking process. The key components include:

1. User Input & File Upload

- Users enter the job description in a text area.
- Resumes are uploaded in PDF format through a file uploader.

2. Text Extraction from Resumes

- PyPDF2 is used to extract text from resumes.
- The extracted text is cleaned by removing special characters, extra spaces, and unnecessary formatting.

3. Preprocessing & Feature Extraction

- Natural Language Processing (NLP) techniques are used:
- Tokenization
- Stop-word removal
- Lowercasing

4. Similarity Computation & Resume Ranking

- Cosine Similarity is used to compare job descriptions and resumes.
- Higher scores indicate a better match between the job description and a resume.
- Resumes are ranked in descending order of similarity.

5. User Interface (UI)

- The Streamlit framework is used to create an interactive web app.
- Results are displayed in a ranked list with similarity scores.
- A data table presents resumes with their match percentage.

4.2 Technologies Used

| Technology | Purpose |
|--------------|--|
| Python | Core programming language |
| Streamlit | UI development for web application |
| PyPDF2 | Extracting text from PDF resumes |
| Scikit-learn | TF-IDF vectorization & Cosine Similarity |
| Pandas | Handling tabular data |
| NumPy | Numerical Computations |

Table.1

4.3 Code Snippets

Extracting Text from Resumes

```
from PyPDF2 import PdfReader

def extract_text_from_pdf(file):
    pdf = PdfReader(file)
    text = ""
    for page in pdf.pages:
        extracted_text = page.extract_text()
        if extracted_text:
            text += extracted_text + " "
    return text
```

Fig.1

Converting Text to TF-IDF Vectors

```
from sklearn.feature_extraction.text import TfidfVectorizer

def vectorize_text(job_description, resumes):
    documents = [job_description] + resumes
    vectorizer = TfidfVectorizer().fit_transform(documents)
    return vectorizer.toarray()
```

Fig.2

Computing Cosine Similarity & Ranking

```
from sklearn.metrics.pairwise import cosine_similarity

def rank_resumes(job_description, resumes):
    vectors = vectorize_text(job_description, resumes)
    job_desc_vector = vectors[0]
    resume_vectors = vectors[1:]

    scores = cosine_similarity([job_desc_vector], resume_vectors).flatten()
    return scores
```

Fig.3

Displaying Results in Streamlit

```
import streamlit as st
import pandas as pd

st.title("AI Resume Screening & Candidate Ranking")

job_description = st.text_area("Enter the job description")
uploaded_files = st.file_uploader("Upload resumes (PDF)", type=["pdf"], accept_multiple_files=True)

if uploaded_files and job_description:
    resumes = [extract_text_from_pdf(file) for file in uploaded_files]
    scores = rank_resumes(job_description, resumes)

    results = sorted(zip([file.name for file in uploaded_files], scores), key=lambda x: x[1], reverse=True)

    df = pd.DataFrame(results, columns=["Resume Name", "Match Score"])
    df["Match Score"] = df["Match Score"].apply(lambda x: round(x * 100, 2)) # Convert to percentage

    st.dataframe(df)
```

Fig.4

4.4 Expected Output

Ranking Resumes

Resume Ranking (Higher is better)

| | Resume Name | Match Score |
|---|--------------------|-------------|
| 0 | Ramu.pdf | 17.68 |
| 1 | RaghuProfile.pdf | 15.89 |
| 2 | Prasad_profile.pdf | 15.13 |

Fig.5

CHAPTER 5

DISCUSSION AND CONCLUSION

5.1 Future Work:

- Integration with ATS for seamless recruitment workflows.
- Multi-format resume support, including DOCX and image-based PDFs.
- Deep learning-based NLP models to improve accuracy.
- Bias detection and mitigation to ensure fair hiring practices.
- Real-time ranking adjustments based on recruiter preferences.

5.2 Conclusion:

The AI-powered Resume Screening and Ranking System significantly improves the efficiency, accuracy, and fairness of the hiring process. By automating resume screening, the system reduces recruiter workload and minimizes human bias in candidate selection. The methodology used in this project ensures a data-driven approach to hiring, improving the quality of shortlisted candidates. Future enhancements, including deep learning integration and ATS compatibility, can further optimize the recruitment process. This project highlights the potential of AI in transforming recruitment workflows and ensuring better hiring decisions.

REFERENCES

1. Goldberg, Y. & Levy, O. (2014). Word2Vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method.
2. Vaswani, A. et al. (2017). Attention Is All You Need. NeurIPS.
3. Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation.