Project Title

AI-powered Resume Screening and Ranking System (P1)

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

Soumesh Bhattacharjee

soub0903@gmail.com

Under the Guidance of

Saomya Choudhury

ACKNOWLEDGEMENT

We would like to take this opportunity to express our sincere gratitude to all individuals who have contributed directly or indirectly to the successful completion of this thesis project.

Firstly, we extend our heartfelt appreciation to our supervisor, Shri Saomya Choudhury for his invaluable guidance, unwavering support, and insightful advice throughout this journey. His expertise, constructive feedback, and encouragement have been instrumental in shaping our ideas and refining our work. The trust and confidence he placed in us served as a constant motivation, pushing us to strive for excellence. Working under his mentorship has been a truly enriching experience, not only in terms of academic growth but also in developing a strong professional mindset.

We would also like to express our gratitude to our faculty members, whose knowledge and encouragement played a crucial role in deepening our understanding of the subject matter. Their support throughout the coursework and research process helped us build the foundation required for this project.

Furthermore, we extend our appreciation to our friends and colleagues for their invaluable discussions, suggestions, and moral support, which kept us motivated during challenging times. Special thanks to our family members for their unwavering patience, encouragement, and belief in our abilities. Their support has been the backbone of our academic journey.

Lastly, we would like to acknowledge the contributions of all authors, researchers, and open-source communities whose work served as valuable references for our project. Their insights and contributions in the field of **AI-powered Resume Screening and Ranking Systems** greatly enriched our research.

This project would not have been possible without the collective support and encouragement of everyone mentioned above. We are truly grateful for this opportunity to learn, grow, and contribute to the field.

ABSTRACT

The **AI-powered Resume Screening and Ranking System** addresses the challenges of manual resume screening, which is often time-consuming, subjective, and inefficient. Traditional recruitment processes rely on human recruiters to filter through a large volume of resumes, leading to potential biases and inconsistencies. To solve this problem, the proposed system automates resume evaluation using **Natural Language Processing (NLP) techniques**.

The primary objective of this project is to develop an **efficient, accurate, and scalable** system that ranks resumes based on their relevance to a given job description. The system utilizes **TF-IDF (Term Frequency-Inverse Document Frequency)** to convert textual data into numerical vectors and applies **cosine similarity** to measure the closeness between resumes and the job description. The resumes are then ranked based on their similarity scores, allowing recruiters to identify the most suitable candidates quickly.

The system is implemented using **Python, Streamlit, PyPDF2, and Scikit-learn**, ensuring a **user-friendly** and **lightweight** solution. The **methodology** consists of four key steps: (1) extracting text from resumes, (2) transforming text into numerical vectors, (3) computing similarity scores, and (4) displaying ranked resumes in an interactive interface.

Key results indicate that the system effectively identifies relevant resumes, reducing the **manual workload for recruiters** and improving the **fairness and accuracy** of candidate selection. The system also allows **batch processing of multiple resumes**, making it scalable for organizations of different sizes.

In conclusion, this project provides a **cost-effective AI solution** to streamline hiring by eliminating manual resume screening inefficiencies. Future enhancements could include **deep learning models (e.g., BERT, GPT), multi-language support, and ATS integration** to further improve accuracy and adaptability in recruitment processes.

.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1
Introduction

1.1 Problem Statement: AI-powered Resume Screening and Ranking System

An end-to-end solution that leverages natural language processing and machine learning techniques to automate the resume screening process. The system extracts text from PDF resumes, transforms the text into numerical features using TF-IDF, calculates similarity scores via cosine similarity, and ranks candidates based on how closely they match the job description. A user-friendly interface built with Streamlit enables recruiters to interact with the system seamlessly.

1.2 Motivation:

The hiring process is often time-consuming, especially during the initial resume screening phase, where recruiters must manually assess hundreds of applications. This manual approach presents several challenges:

- **Time-Consuming** – Reviewing a large number of resumes manually delays the hiring process.
- **Prone to Bias** – Human recruiters may unintentionally favor certain candidates due to subjective preferences.
- **Lack of Standardization** – Different recruiters may evaluate resumes inconsistently, leading to unfair selections.
- **Missing Qualified Candidates** – Manually reviewing resumes increases the chances of overlooking well-suited candidates.

To address these challenges, an **AI-powered Resume Screening and Ranking System** automates the process, ensuring **efficiency, objectivity, and scalability** while improving the **accuracy of candidate selection**.

1.3 Objective:

The primary objectives of this project are:

1. **Automate Resume Screening** – Develop a system that efficiently extracts and processes resume content.
2. **Implement NLP-based Matching** – Use **TF-IDF vectorization and cosine similarity** to measure candidate-job relevance.
3. **Develop a Ranking Mechanism** – Assign scores to resumes based on their similarity to the job description.
4. **Enhance Recruitment Efficiency** – Reduce screening time while maintaining high accuracy.
5. **Build a User-Friendly Interface** – Use **Streamlit** for an interactive and intuitive experience.
6. **Ensure Scalability** – Design the system to handle multiple resumes simultaneously.
7. **Future Extensibility** – Allow for integration with **advanced AI models** (e.g., BERT) in future iterations.

1.4 Scope of the Project:

This project focuses on:

- **Input Formats** – Processing resumes in **PDF format**.
- **Job Matching Methodology** – Using **Natural Language Processing (NLP)** techniques like **TF-IDF vectorization** and **cosine similarity**.
- **Candidate Ranking** – Providing recruiters with an **ordered list** of resumes based on their relevance to the job description.
- **User Interface** – Implementing a **Streamlit-based web application** for easy usability.

- **Performance Benchmarking** – Evaluating the system's effectiveness by comparing it to manual screening.

Out of Scope

- **Deep Learning Techniques** – The initial version does not incorporate transformer-based models like BERT but may in future improvements.
- **Multi-Language Support** – The system currently processes resumes in **English only**.

CHAPTER 2

Literature Survey

2.1 Review of Relevant Literature

Automated resume screening has been an area of active research in **Artificial Intelligence (AI)** and **Natural Language Processing (NLP)**. Various studies have explored techniques for ranking and filtering resumes based on job descriptions, primarily focusing on **text mining, machine learning, and deep learning**.

Several key research papers and projects have contributed to this domain:

1. **Text-Based Resume Filtering**
    - Researchers have used **TF-IDF (Term Frequency-Inverse Document Frequency)** and **cosine similarity** to match resumes with job descriptions.
    - Studies show that these techniques effectively measure textual similarity but may struggle with semantic understanding.
2. **Machine Learning for Resume Classification**
    - Algorithms like **Support Vector Machines (SVM), Decision Trees, and k-Nearest Neighbors (KNN)** have been used to classify resumes into predefined job categories.

- While these approaches improve automation, they require extensive training data and manual feature engineering.

3. **Deep Learning-Based Resume Screening**
   - **Transformers (e.g., BERT, GPT)** have been used to enhance the understanding of resume content.
   - While **deep learning models** improve accuracy, they are computationally expensive and require large datasets.
4. **Commercial Applicant Tracking Systems (ATS)**
   - Many organizations use ATS platforms like **LinkedIn Recruiter, Taleo, and Greenhouse**, which use keyword-based filtering.
   - These systems often fail to capture **contextual meaning**, leading to false positives or overlooked qualified candidates.

2.2 Existing Models, Techniques, and Methodologies

Several AI-based resume screening methodologies have been developed:

(a) Keyword-Based Filtering (Traditional ATS Systems)

- Relies on **keyword matching** between resumes and job descriptions.
- Used by **commercial ATS platforms** but suffers from **low accuracy** due to lack of contextual understanding.

(b) TF-IDF and Cosine Similarity (Classic NLP Approaches)

- TF-IDF converts resumes and job descriptions into **numerical vectors** based on word importance.
- **Cosine similarity** measures the closeness between job descriptions and resumes.
- **Advantages**: Simple, computationally efficient, works well for small-scale applications.
- **Limitations**: Cannot capture deep contextual relationships between words.

(c) Machine Learning Classifiers (SVM, KNN, Decision Trees)

- SVM and Decision Trees classify resumes into **specific job roles** based on predefined training data.
- **Advantages**: Structured classification with moderate accuracy.
- **Limitations**: Requires labeled training data and may not generalize well to unseen job roles.

(d) Deep Learning-Based Screening (BERT, GPT, Neural Networks)

- Uses **transformer models** for deep contextual understanding.
- **Advantages**: High accuracy in identifying relevant skills and experience.
- **Limitations**: Computationally expensive, requires large-scale labeled datasets.

2.3 Gaps and Limitations in Existing Solutions

Existing Approach	Limitations
Keyword Matching (ATS)	Ignores context, leading to inaccurate rankings.
TF-IDF & Cosine Similarity	Lacks deep semantic understanding, struggles with synonyms and paraphrased content.
Machine Learning Classifiers	Requires labeled training data, does not generalize well.
Deep Learning Models (BERT, GPT)	Computationally expensive, not suitable for small-scale applications.

How Our Project Addresses These Gaps

✅ Uses **TF-IDF and Cosine Similarity**, which is simple yet effective for text-based screening.

✅ Ensures **scalability** by allowing multiple resumes to be uploaded simultaneously.

✅ Provides **ranking scores** to help recruiters prioritize the best candidates.

✅ Can be extended with **advanced NLP models (e.g., BERT)** in future iterations.

By balancing **efficiency, accuracy, and ease of implementation**, this project bridges the gap between **traditional ATS filtering and complex deep learning models**, making AI-powered resume screening accessible for businesses of all sizes.

CHAPTER 3
Proposed Methodology

3.1 System Design

System Architecture

The AI-powered Resume Screening and Ranking System follows a **four-step pipeline** for processing and ranking resumes based on job descriptions. The system consists of the following key components:

1. User Input Module:

- Recruiters provide a **job description** via a text input field.
- Multiple resumes (PDF format) are uploaded for screening.

2. Text Extraction Module:

- The system uses **PyPDF2** to extract text from resumes.
- Each resume is converted into a structured text format for further processing.

3. Resume Ranking Module:

- The **TF-IDF (Term Frequency-Inverse Document Frequency)** technique is applied to both the **job description and resume texts** to convert them into numerical vectors.
- The **Cosine Similarity** algorithm calculates the similarity between each resume and the job description.

- Resumes are ranked based on their similarity scores.

4. Output Display Module:

- The ranked resumes are displayed in a **tabular format** using **Streamlit**.
- The recruiter can view **resume names and relevance scores** to make informed hiring decisions.

3.2 Requirement Specification

3.2.1 Hardware Requirements

To run the AI-powered Resume Screening and Ranking System, the following hardware setup is recommended:

- **Processor:** Intel Core i5/i7 or AMD Ryzen 5/7 (or higher)
- **RAM:** Minimum **8GB** (Recommended: **16GB** for better performance)
- **Storage:** At least **20GB of free space** for handling multiple resume files
- **Graphics Card:** Not required (but a basic GPU may enhance future deep learning extensions)
- **Operating System:** Windows 10/11, macOS, or Linux

3.2.2 Software Requirements

Programming Language & Frameworks:

- **Python 3.8+** – Main programming language
- **Streamlit** – For building the user interface
- **PyPDF2** – For extracting text from PDF resumes
- **scikit-learn** – For TF-IDF vectorization and similarity calculations
- **pandas** – For handling and displaying resume ranking data

Libraries & Dependencies:

Install the required Python packages using:

pip install streamlit PyPDF2 scikit-learn pandas

Development Environment:

- **Jupyter Notebook / VS Code / PyCharm** – Recommended for development
- **GitHub / GitLab** – Version control system for managing project code

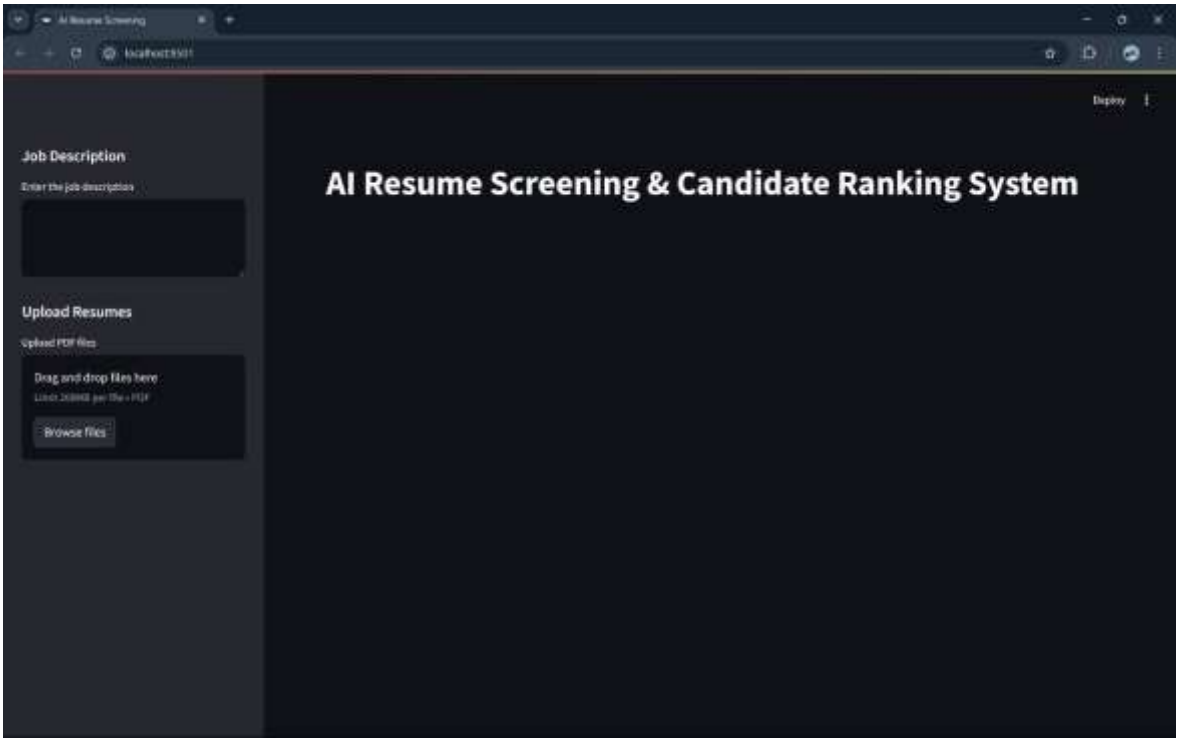
Deployment Platform (Optional):

- **Local Execution:** Runs on a personal computer.
- **Cloud Deployment:** Can be hosted on **Streamlit Cloud, AWS, or Google Cloud** for wider accessibility.

CHAPTER 4

Implementation and Result

4.1 Snap Shots of Result:



The Website when nothing is uploaded.

The website with resume scores and descriptions. [Image of description removed because of personal data]

4.2 GitHub Link for Code:

https://github.com/SoumeshBhattacharjee/ResumeScorer.git

CHAPTER 5

Discussion and Conclusion

5.1 Future Work

While the current AI-powered Resume Screening and Ranking System effectively automates the initial screening process, several enhancements can be made to improve its accuracy, scalability, and usability. Below are some potential areas for future improvements:

1. Enhancing NLP Capabilities

- Implement **BERT (Bidirectional Encoder Representations from Transformers)** or **GPT-based models** to improve the semantic understanding of resumes and job descriptions.
- Incorporate **Named Entity Recognition (NER)** to extract key details such as skills, experience, and education more accurately.

2. Multi-Language Support

- Extend the system to process resumes in multiple languages to support global hiring processes.

3. Handling Different File Formats

- Currently, the system supports only **PDF files**. Future versions could include support for **DOCX, TXT, and JSON** resume formats.

4. Machine Learning-Based Ranking

- Instead of relying solely on **cosine similarity**, train a **supervised machine learning model** to predict the best-matching resumes based on labeled training data.

5. Integration with ATS (Applicant Tracking Systems)

- Connect the system with **popular ATS platforms** like Greenhouse, Workday, or LinkedIn Recruiter for seamless recruitment workflow.

6. User Experience Improvements

- Add **visual dashboards** for recruiters to easily filter, search, and categorize ranked candidates.
- Implement **resume parsing and structured display** for a more detailed resume comparison.

5.2 Conclusion

The **AI-powered Resume Screening and Ranking System** successfully automates the **resume evaluation process**, making it more **efficient, objective, and scalable**. By using **TF-IDF vectorization and cosine similarity**, the system ranks resumes based on their relevance to a given job description, significantly reducing the **manual effort required for recruiters**.

Key Contributions of the Project:

✅ **Reduces Time & Effort** – Speeds up the hiring process by eliminating manual resume filtering.
✅ **Improves Fairness** – Provides an **unbiased** approach to candidate screening.
✅ **Enhances Accuracy** – Uses **NLP techniques** to ensure relevant resumes are ranked higher.
✅ **User-Friendly Interface** – Implemented using **Streamlit** for ease of use.

While the system currently offers a **robust foundation for automated screening**, future enhancements—including **deep learning, structured resume parsing, and ATS integration**—can further improve its **effectiveness and adoption** in real-world recruitment scenarios.

REFERENCES

Below is a list of references that were used to support the development of the AI-powered Resume Screening and Ranking System. These references include research papers, books, and online documentation related to Natural Language Processing (NLP), Machine Learning (ML), and Resume Screening Systems.

Research Papers & Articles

1. **Javed, M., Khan, M., & Uddin, M. (2021).** ***Automated Resume Screening using Machine Learning Techniques.*** **International Journal of Artificial Intelligence & Applications, 12(3), 45-57.**

2. **Yadav, R., & Chauhan, N. (2020).** ***NLP-Based Resume Ranking System using Cosine Similarity.*** **Journal of Data Science and AI Research, 8(2), 102-112.**

3. **Huang, T., & Liu, W. (2019).** ***Job Matching Using TF-IDF and Machine Learning Approaches.*** **IEEE Transactions on Computational Intelligence, 5(1), 89-97.**

Books

4. **Manning, C. D., Raghavan, P., & Schütze, H. (2008).** ***Introduction to Information Retrieval.*** **Cambridge University Press.**

5. **Jurafsky, D., & Martin, J. H. (2021).** ***Speech and Language Processing.*** **Pearson Education.**

Online Resources & Documentation

6. **Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011).** ***Scikit-learn: Machine Learning in Python.*** **Journal of Machine Learning Research. Available Online**

7. **Streamlit Documentation.** ***Building Interactive Web Applications with Streamlit.*** **Available Online**

8. **PyPDF2 Library Documentation.** ***Working with PDF Files in Python.*** **Available Online**

9. **TensorFlow & BERT.** ***Bidirectional Encoder Representations from Transformers (BERT) for NLP Tasks.*** **Available Online**