Web Application for Resume Optimization and Applicant Tracking System using ML

Mini Project for Semester 6

Department of Information Technology

B.Tech (Information Technology)

Team Members:

Apoorv Mishra (16010422073)
Aarush Saboo (16010422097)
Ojas Keer (16010422054)
Heramb Patil (16010422088)

Abstract

This project presents a web application designed to optimize resumes for Applicant Tracking Systems (ATS) using Natural Language Processing (NLP). The application evaluates resumes based on top-tier ATS standards and calculates an ATS score using a pre-trained supervised machine learning model and TF-IDF vectorization. Additionally, the app provides improvement suggestions using T5 transformers, offering real-time feedback for better job matching.

Furthermore, the project features a job description matching tool, allowing users to paste their job description (JD) and compare it with their uploaded resume. The system evaluates the compatibility between the JD and the resume.

Users can upload resumes, input job descriptions, and receive personalized optimization recommendations, enhancing their chances of being shortlisted by automated recruitment systems.

Keywords**:** Applicant Tracking System, TF-IDF, Vectorization, Supervised learning and text parsing

Contents

1 Introduction……... **iv**

1.1 Problem Definition…….. 4

1.2 Motivation …….. 4

1.3 Scope of Project and Objectives..……. 4

1.4 Functional and Non-Functional Requirements …….. 5

1.5 Organization of the Report …….. 5

2 Background Work……. vi

2.1 Overview of Technologies…… 6

3 Literature Survey…… …….. **viii**

4 Project Plan and Timeline…… ix

5 Implementation……... xi

6 Results and Discussions……. xiii

7 Conclusion and Future Work……. xiv

References ……... **XV**

Acknowledgements……. XVI

Chapter 1

Introduction

1.1 Problem Definition

An increasing need for ATS optimization in job applications and the difficulty in manually ensuring that resumes are ATS-friendly.

1.2 Motivation

The rise of automated recruitment systems has necessitated optimized resume formats, and this project aims to automate that process using machine learning. Overall, the following points state our motivation to build this project:

- **Rising Job Market Competition**: With more job applicants, ATS is increasingly used to screen resumes, making it harder for job seekers to stand out.
- **Time-Consuming Process**: Optimizing a resume for ATS requires manual effort, expertise, and time, which many job seekers lack.
- **Improving Job Seekers' Chances**: The project aims to help job seekers increase their chances of getting noticed by automating resume optimization and providing real-time feedback.

1.3 Scope and Objectives

The objective of this project is to develop a web application powered by Machine Learning for resume optimization and ATS compatibility checking. The platform will provide job seekers with suggestions on improving their resumes, ensuring that they are ATS-friendly. The system will:

- Optimize resumes based on ATS criteria.
- Offer suggestions for improving resume formatting and keyword usage.
- Compares the user's resume with the provided Job Description (JD) and assesses how well the resume aligns with the job requirements.

1.4 Functional and Non-Functional Requirements

Functional Requirements:

- Resume Parsing: Extracts and processes resume data from PDF, DOCX, and TXT formats.
- ATS Compatibility Checker: Analyses the resume for ATS compatibility based on keyword density, section headings, and formatting.
- Resume Optimization: Provides Suggestions for resume improvements (e.g., keyword enhancements, grammar corrections).

Non Functional Requirements:

- Performance: The system should process resumes quickly, providing feedback in under 5 minutes.
- Usability: The platform must be user-friendly, with an intuitive interface.
- Scalability: The system should handle a large volume of resumes and job applications without performance degradation.

1.5 Organization of the Report

This report is structured as follows:

Chapter 2: Literature Survey - Overview of existing methods for resume optimization and ATS compatibility.

Chapter 3: Project Plan and Timeline - Details on the project phases and timeline.

Chapter 4: Implementation - Describes the system architecture, methodology, and technologies used.

Chapter 5: Results and Discussion - Analysis of system performance and ATS compatibility testing.

Chapter 6: Conclusion and Future Work - Summarizes the findings and proposes future enhancements.

Chapter 2

Background Work

2.1 Overview of Technologies

The development of the web-based Resume Optimization and ATS Compatibility System leverages key technologies of Machine Learning (ML), and web development tools. These technologies help to align resume with job descriptions and improve ATS compatibility.

1. Natural Language Processing (NLP)

SpaCy: Extracts structured data such as skills, job titles, and company names from resumes

T5 Transformer: Optimizes resume content by suggesting improvements.

Pdfplumber: Library for extracting text from PDF resumes.

TF-IDF: This helps in keyword matching between resumes and job descriptions.

Sentence-BERT: Used for comparing resumes to job descriptions and calculating how well they align based on semantic similarity.

Supervised Learning: Used for predicting ATS compatibility based on labeled data**.**

2. ATS Compatibility Scoring

The system evaluates resumes for ATS compatibility by checking:

Keyword Relevance: Using TF-IDF to identify and match key terms.

Formatting: Ensures resumes are structured to pass ATS filters.

3. Frontend Development

ReactJS: Provides an interactive user interface for uploading and editing resumes with real-time feedback.

CSS: Enhance UI design for a smooth user experience.

5. Backend Development

Node.js & Express.js: Handle API requests, resume parsing, and job description comparison.

MongoDB: Stores resume data and user profiles.

Chapter 3

Literature Survey

Research Area	Methodology	Findings/Improvements	Reference
Resume Parsing with Deep Learning Models	Deep learning models, including CNN and Bi-LSTM, were used for more efficient resume parsing, achieving higher accuracy in extracting segmented information.	Convolutional Neural Networks (CNN), Bi-LSTM, CRF	Ayishathahira C H, Sreejith C, Raseek C. "Combination of Neural Networks [Link]
Resume & Job Description Matching	Proposed transformer-based models for predicting resume suitability for a given job, achieving high classification accuracy and outperforming traditional models.	Transformer-based classification, Multi-head attention	Changmao Li, Elaine Fisher, Rebecca Thomas, Steve Pittard, Vicki Hertzberg, Jinho D. Choi
Competence-Level Prediction in Job Matching	Introduced transformer models for classifying resumes by competence level, leading to improved prediction of suitable job levels for candidates.	Transformer encoder, section encoding, attention	Changmao Li, Elaine Fisher, Rebecca Thomas, Steve Pittard, Vicki Hertzberg, Jinho D. Choi

Chapter 4

Project Plan and Timeline

The project was executed over a structured timeline, spanning from **January** and continuing until **April**. Well-defined phases to ensure systematic progress from ideation to final deployment. Below is an overview of each phase:

Phase 1: Ideation and Research (January – Mid February)

Activities:

- Finalized the project concept focused on web-based resume optimization using ML
- Finalized team formation and division of responsibilities.
- Designed the initial architecture, including frontend, backend, and database stack.
- Choose the tech stack (e.g., React/**Streamlit**, MongoDB, NodeJs, TF-IDF).

Outcome: A clear project scope and an architecture plan with selected tools and defined responsibilities.

Phase 2: Dataset Collection and Preliminary Research (Mid February to March)

Activities:

- Conducted extensive research on existing ATS systems and AI-based resume scoring
- Collected benchmark datasets (Princeton Resume Dataset) and reviewed formatting guidelines from MIT, Harvard, UPenn, etc.
- Defined functional and non-functional requirements for the system.

Outcome: Well-defined scope, identified key modules (resume parsing, scoring, editing), and established feasibility.

Phase 3: Training of Model using the database (Mid-March- April)

Activities:

- Implemented NLP based Resume parsing using *pdfplumber and SpaCy*
- Used High Quality labelled resumes to train the optimizer
- Fine-tuned scoring algorithms based on section headings, grammar, keyword density

Outcome: Base Model for for resume analysis, grammar suggestion, keyword matching, and ATS score prediction is functional

Phase 4: Frontend Development (April)

Activities:

- Developed an interactive front end using ReactJS.
- Enabled real-time resume editing with inline feedback for improvements.
- Designed user-friendly UI with chart components to visualize ATS scores and keyword density.

Outcome: Functional and aesthetically clean frontend with integration-ready components

Phase 5: Integration, Analytics & API Work (Upcoming - Late April)

Planned Activities:

- Integrate frontend and backend APIs.
- Implement job description to resume matching module (Sentence-BERT).
- Build a dashboard to show performance metrics, improvement suggestions, and resume health.
- Enable PDF download and grammar-corrected resume export.

Expected Outcome: A fully functional resume optimization platform with editing, scoring, and job matching features.

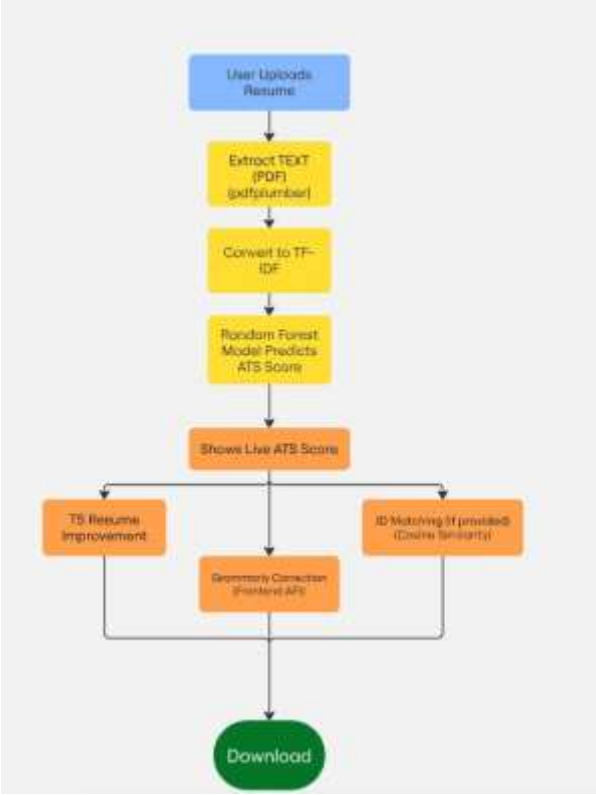
Chapter 5

Implementation

5.1 System Architecture

The system is divided into two main components:

- **Frontend:** Built using ReactJS, the frontend allows users to upload their resumes and receive real-time feedback on ATS compatibility. It includes:
    - **Resume Upload:** Users can upload resumes in PDF, DOCX, or TXT format.
    - **ATS Feedback:** The system provides real-time suggestions for improving the resume.
- **Backend:** Built using Node.js & Express.js, the backend handles:
    - **Resume Parsing:** Using pdfplumber, it processes and extracts text from resumes.
    - **ATS Compatibility** Checking: The backend compares resume content with job descriptions, providing ATS compatibility scores.
    - **Resume Optimization**: Uses T5 Transformer to suggest keyword optimizations and formatting changes.



5.2 Algorithm and Methodology

The system follows the following steps:

1. **Resume Parsing:** The resume is uploaded, and its content is extracted using pdfplumber.
2. **ATS Compatibility Analysis:** The resume is analyzed for keyword density and section headings. TF-IDF is used to match relevant terms with the job description.
3. **Resume Optimization:** T5 Transformer provides suggestions to improve keyword usage, grammar, and resume structure.
4. **Job-Resume Comparison:** Sentence-BERT measures how closely the resume matches the job description.

5.3 Technologies and Libraries Used

- **Frontend**: ReactJS, CSS for styling**.**
- **Backend:** Node.js, Express.js, MongoDB.
- **NLP:** SpaCy, T5 Transformer, TF-IDF, Sentence-BERT**.**
- **Resume Parsing:** pdfplumber**.**
- **Dataset**: Initially the Model was trained for Good resumes, which were taken from the students of Princeton University

Chapter 6:

Results and Discussion

6.1 Testing and Evaluation

The system was tested with a variety of resumes and job descriptions to evaluate its performance:

- **ATS Compatibility:** The system accurately analyzed resumes and identified key areas for improvement, such as missing keywords and improper formatting.
- **Keyword Optimization:** Suggestions for keyword usage aligned with industry-specific terminology and job descriptions.
- **Job-Resume Matching:** The Sentence-BERT model provided high accuracy in assessing how well the resume matched the job description, with results improving after suggestions were implemented.

6.2 Performance

- **Speed:** The system processed resumes in under 3 minutes, well within the expected performance limits.
- **Usability:** The frontend was responsive, with real-time feedback provided for users.

Chapter 7

Conclusion and Future Work

7.1 Conclusions

The Web-Based Resume Optimization and ATS Compatibility System successfully automates the process of optimizing resumes for ATS systems. By analyzing resumes for keyword relevance, formatting, and structural integrity, the system helps job seekers improve their chances of getting noticed by automated recruitment systems.

7.2 Future Enhancements

Future work may include:

- Real-Time Resume Evaluation: Implementing real-time optimization suggestions as users' type.
- Enhancing the Frontend and Web application: Further work is required to develop the Web Application of this Program
- Model Training: The model created was trained on a limited dataset and needs more resumes to improve accuracy.

References

[1] M. Peicheva, “Data Analysis from Applicant Tracking Systems,” 2023.

[2] MIT Sloan, “Job Seekers with AI-Boosted Resumes More Likely to Be Hired,” 2023.

[3] Emory University, “CRC-Competence-Level Prediction and Resume Matching,” [Online].

[4] IOP Conference, “Resume Extraction with Conditional Random Fields (CRF),” 2020.

[5] Unknown Author, “Combination of Neural Networks & CRF for Resume Parsing,” [Online].

[6] Unknown Author, “A Machine Learning-Based AI Framework to Optimize Recruitment Screening Process,” 2024.

[7] Kent State University, “Research on ATS Optimization & Hiring Trends,” [Online].

[8] University of Pennsylvania Career Services, “Optimizing Your Resume for AI Scanners,” 2024.

Acknowledgements:

We sincerely thank our guide, Prof. Ravindra Divekar, for his valuable guidance and support throughout this project. We also thank the Department of Information Technology for providing the necessary resources and encouragement.

We appreciate the efforts of our team members — Apoorv Mishra, Heramb Patil, Aarush Saboo, and Ojas Keer — for their collaboration and dedication. This project was a great learning experience for all of us.