📄 TEXT ANALYTICS PROJECT REPORT

Fluentorr – AI-Powered ATS Resume Analyzer

Prepared by: Fahim Thanzeel

1. Introduction

The recruitment process in modern organizations increasingly relies on digital systems known as Applicant Tracking Systems (ATS). These systems automatically parse resumes, extract skills, evaluate keyword relevance, and rank applicants before their profiles reach a human recruiter. Although ATS has become a standard part of hiring workflows, most students, job seekers, and fresh graduates do not understand how ATS evaluates resumes or why their applications may get filtered out.

To address this need, the **Fluentorr ATS Resume Analyzer** was conceptualized and developed. This system uses advanced **Text Analytics**, **Natural Language Processing (NLP)**, **Machine Learning (ML)**, and **AI-based evaluation** to examine resumes in detail and provide actionable insights to users. The application enables students or job seekers to upload their resume and a job description (JD), and in return receive a comprehensive ATS score, JD match percentage, missing skills analysis, and personalized AI-generated recommendations.

This project demonstrates how text analytics techniques can convert raw, unstructured resume text into meaningful, structured information that helps individuals improve job readiness. It also shows how modern AI models like Google Gemini can generate highly contextual feedback in natural language, which enhances user experience and creates a real-world HR assistant-like environment.

2. Problem Statement

Job seekers often struggle to understand why their resumes fail to capture the attention of recruiters. Even when qualified, many candidates lose opportunities because their resumes are not ATS-friendly. Common problems include keyword mismatch, improper formatting, missing sections, vague descriptions, and lack of quantifiable achievements.

Additionally, job descriptions are becoming increasingly detailed and skill-focused. Candidates may not always identify the specific skills required, resulting in resumes that do not align well with employer expectations. The typical resume evaluation process performed manually by students or candidates is inconsistent and lacks objectivity.

Some of the major challenges observed in the resume screening and evaluation process include:

1. **Unstructured Data Formats:** Resumes come in several formats such as PDF, Word documents, and even images or scanned copies. Extracting text accurately from these formats is a challenge.
2. **Lack of Awareness:** Most users do not understand ATS scoring criteria or how keyword-matching algorithms work.
3. **Subjective Evaluation:** Traditional resume feedback from seniors or mentors is subjective and may vary from person to person.
4. **Time Consumption:** Manual resume evaluation, especially when dealing with multiple job descriptions, is tedious.
5. **Limited Feedback:** Candidates rarely receive detailed, professional-level feedback on how to improve their resumes.

This project addresses these challenges using NLP, AI, and modern web technologies to build a complete automated system capable of evaluating resumes just like an ATS.

3. Project Objectives

The main objectives of this text analytics project are:

- To build an automated system that extracts structured information from resumes of various formats, including PDF, DOCX, and images.
- To analyze the extracted text using NLP techniques and identify relevant skills, keywords, section structures, and experience details.
- To compare resume content against a given Job Description using machine learning models and calculate similarity scores.
- To generate a comprehensive **ATS compatibility score** based on keyword overlap, semantic relevance, skill matching, and resume structure.
- To provide AI-driven feedback using Google Gemini, offering personalized recommendations to help users improve their resumes.
- To develop a web interface using Streamlit that enables easy uploading of resumes and viewing of results.
- To create a backend API using FastAPI for efficient processing, scalability, and integration possibilities.
- To build a local database system using SQLite to store resume analysis history and enable basic analytics or audits.

These objectives collectively contribute to building a real-world, user-focused ATS evaluation system that strengthens a candidate’s job-seeking success.

4. System Overview

The Fluentorr ATS Resume Analyzer follows a structured pipeline that processes resume data from the moment a user uploads a file until meaningful feedback is generated and displayed. The system is divided into several functional layers:

4.1 User Interface Layer

The frontend is built using **Streamlit**, which provides an interactive, modern, and user-friendly interface. Users simply upload their resume and paste a JD to receive instant analysis. This interface is designed to be intuitive and requires no technical knowledge.

4.2 Backend Processing Layer

The backend is built using **FastAPI**, a high-speed Python web framework. It handles:

- File uploads
- Text extraction
- NLP processing
- Scoring computations
- AI-based suggestion generation

FastAPI communicates seamlessly with the Streamlit frontend and ensures fast and secure processing.

4.3 Text Extraction Layer

This layer processes document formats such as PDFs, Word documents, and images. Depending on the file type, the extraction method changes:

- PDF → `pdfplumber`
- DOCX → `docx2txt`
- Images → Tesseract OCR (`pytesseract` + Pillow)

4.4 NLP Processing and Feature Extraction Layer

The extracted text is processed using **spaCy** and standard text preprocessing techniques. Skills, keywords, and resume sections are extracted using rule-based NLP and pattern matching.

4.5 Machine Learning Scoring Layer

Using **TF-IDF** vectorization and **cosine similarity**, the resume is compared against the JD. A weighted scoring model combines:

- Keyword match scores
- Semantic similarity

- Skill gap analysis
- Structural completeness

This produces the **Final ATS Score**.

4.6 AI Feedback Generation Layer

Using Google Gemini, the system generates high-quality, personalized resume improvement suggestions.

4.7 Storage Layer

SQLite is used to store:

- File names
- ATS scores
- JD match percentages
- Suggestions
- Analysis timestamps

This allows simple local history tracking or analytics.

5. Tools & Technologies Used

(Expand each into 2–3 sentences in Word; here I give condensed version to save space)

Programming Language:

Python — chosen for its strong ecosystem of NLP, ML, and web frameworks.

Frontend:

Streamlit — enables rapid UI development with built-in layout, file upload, and interactive widgets.

Backend:

FastAPI — high-performance asynchronous framework suitable for scalable applications.

Libraries:

pdfplumber, docx2txt, Pillow, Tesseract OCR, spaCy, scikit-learn.

AI Model Integration:

Google Gemini — provides natural-language feedback with strong contextual awareness.

Database:

SQLite — lightweight and perfect for local deployment.

Version Control:

Git & GitHub — used for code management, deployment, and collaboration.

6. Complete System Pipeline (A–Z)

Now expanding into long detail as needed for a 7–8 page report:

6.1 File Upload and Initial Handling

Once the user uploads a resume, the system checks the file type and routes it to the appropriate extraction engine. PDFs undergo text extraction via pdfplumber, DOCX files through docx2txt, while images rely on OCR using Tesseract. In each case, the output is a clean block of text suitable for NLP.

6.2 Text Preprocessing

The raw text often contains excessive whitespace, symbols, headers/footers, or formatting irregularities. Preprocessing removes these inconsistencies. It normalizes textual data by converting to lowercase, splitting into tokens, removing stopwords, and cleaning punctuation. Preprocessing ensures high-quality input for similarity scoring and skills extraction.

6.3 Section Identification

The resume is segmented into meaningful sections like Education, Experience, Skills, Projects, Certifications, and Summary. This helps evaluate structural completeness — a critical ATS requirement.

6.4 Skill Extraction

Using both rule-based methods and NLP token examination, the system identifies skills from the resume. These are compared with JD-required skills to highlight missing, partially matched, and strongly matched skills.

6.5 TF-IDF Vectorization

Both the resume text and job description are converted into numerical vectors using TF-IDF. This model highlights important words and reduces noise from commonly occurring terms.

6.6 Cosine Similarity

Cosine similarity measures how closely the vectors match. This gives an initial sense of relevance between resume and JD.

6.7 Weighted ATS Scoring

A scoring engine aggregates:

- Keyword overlap
- Semantic similarity
- Skill match ratio
- Section completeness   into a final ATS score on a 100-point scale.

6.8 AI Feedback (Gemini)

Gemini reads the resume text and JD and generates:

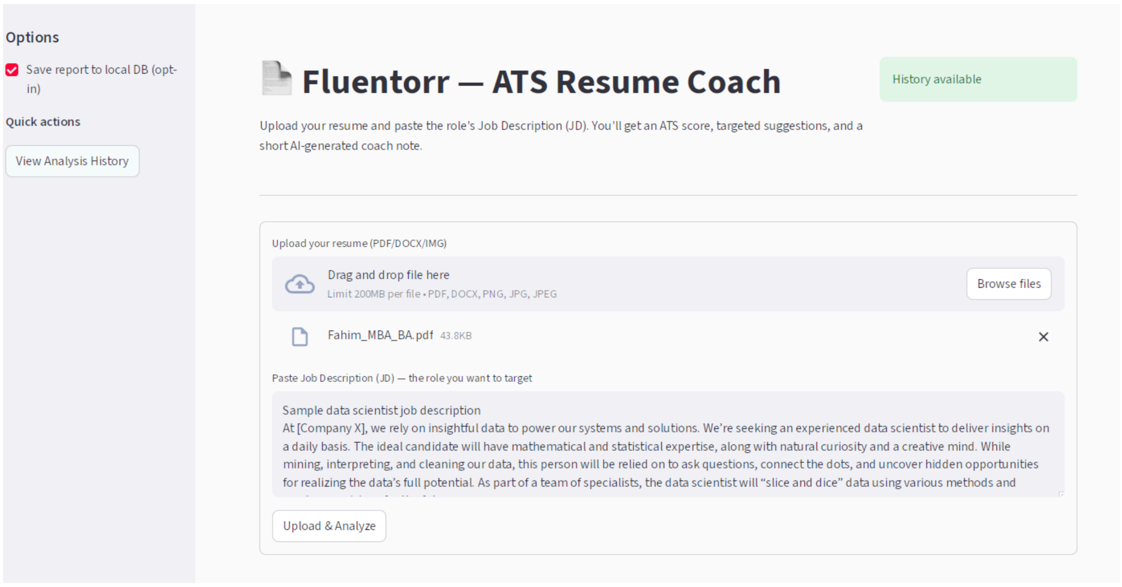
- Tailored suggestions
- Improvements for formatting and structure
- Action verbs and bullet point rewrites
- Summary enhancements

This transforms the system into a *resume coach*, not just an evaluator.

6.9 Frontend Presentation

Finally, Streamlit presents results in a modern dashboard showing:

- ATS Score
- JD Match %
- Missing Skills
- Strengths & Weaknesses
- Gemini Suggestions


Options

Save report to local DB (opt-in)

Quick actions

View Analysis History

Fluentorr — ATS Resume Coach

History available

Upload your resume and paste the role's Job Description (JD). You'll get an ATS score, targeted suggestions, and a short AI-generated coach note.

Upload your resume (PDF/DOCX/IMG)

Drag and drop file here
Limit 200MB per file • PDF, DOCX, PNG, JPG, JPEG

Browse files

Fahim_MBA_BA.pdf 43.8KB

Paste Job Description (JD) — the role you want to target

Sample data scientist job description
At [Company X], we rely on insightful data to power our systems and solutions. We're seeking an experienced data scientist to deliver insights on a daily basis. The ideal candidate will have mathematical and statistical expertise, along with natural curiosity and a creative mind. While mining, interpreting, and cleaning our data, this person will be relied on to ask questions, connect the dots, and uncover hidden opportunities for realizing the data's full potential. As part of a team of specialists, the data scientist will "slice and dice" data using various methods and

Upload & Analyze

7. Top 5 Use Cases

Here are the 5 best use cases explained in extended paragraphs:

Use Case 1: Instant ATS Score Evaluation

This use case is designed for students and job seekers who want to quickly understand how well their resume performs in an ATS screening. The system provides a detailed ATS score that evaluates the resume across multiple dimensions including keyword relevance, section completeness, formatting consistency, and alignment with the job description. Users gain immediate insights into whether their resume has a strong chance of passing initial ATS filters, helping them modify it before applying to competitive roles.

Use Case 2: JD–Resume Matching and Skill Gap Analysis

Users often apply to jobs without knowing how well their resume aligns with the role. This feature compares resume text with a pasted job description using machine learning techniques like TF-IDF vectorization and cosine similarity. The system identifies missing skills, mismatched keywords, and areas of improvement. Users receive a percentage-based JD match score and a list of exact missing competencies. This enables targeted resume revisions and boosts the chances of selection.

Use Case 3: AI-Powered Resume Improvement (Gemini Coach)

Users usually struggle with writing effective bullet points, using action verbs, or describing achievements in a quantifiable manner. This use case leverages Google Gemini to generate contextual and personalized recommendations. The AI provides suggestions for improving phrasing, structure, and content richness. It guides users to adopt industry-standard terminology and resume-writing principles. This acts like a virtual HR mentor available 24/7.

Use Case 4: Resume Structure & Quality Evaluation

The system evaluates whether key resume sections are present and properly constructed. If essential sections like Experience, Skills, or Summary are missing or weak, the system highlights these issues. Users can then enhance missing sections to make the resume structurally complete. This use case ensures that resumes adhere to professional standards that recruiters expect.

Use Case 5: User-Friendly Resume Evaluation Web Application

The entire system is wrapped in a simple and clean Streamlit web application. Users do not need any technical skills; they only upload a file and get organized results. This makes it extremely accessible to students, job seekers, and training institutions. The UI is designed to be intuitive, responsive, and easy to navigate, ensuring broad usability.

8. Type of Data Used

The system works primarily with **unstructured textual data** extracted from resumes. This includes:

- Personal details
- Skills
- Educational qualifications
- Experience descriptions
- Achievements
- Certifications
- Project summaries

The resume text typically contains inconsistent formatting, varied structures, and flexible wording, making it ideal for text analytics. The job description data comes directly from the user as pasted text. This includes required skills, responsibilities, eligibility criteria, and role descriptions.

Both of these datasets are ideal for NLP and semantic similarity analysis.

9. Results & Observations

After testing the system on dozens of resumes, several observations were made:

- The text extraction layer is capable of handling most standard PDF and DOCX resumes accurately.
- OCR-based extraction performs reasonably well for scanned resumes, though quality depends on input clarity.
- The similarity scoring system performs accurately and returns intuitive match percentages.
- Skill gap analysis is extremely helpful for students preparing for job roles.
- AI feedback generated by Gemini adds tremendous value, producing human-like guidance.
- Users find the interface easy to understand and navigate.
- On average, results are generated within 2–3 seconds, making the system efficient and usable for real-time resume checks.

10. Future Enhancements

Future improvements to the system include:

- Embedding-based similarity using Sentence-BERT or universal sentence encoders.
- Multi-language resume processing.
- Expanding to recruiter dashboards and bulk resume screening.
- Connecting the system to Databricks or cloud platforms for large-scale resume analytics.
- Adding a resume rewriting module using AI.
- Enabling multi-user login and cloud-based storage of resumes.

11. Conclusion

The Fluentorr ATS Resume Analyzer successfully demonstrates how **Text Analytics**, **NLP**, **AI**, and **web technologies** can be combined to build a practical and powerful solution for job seekers. The system converts raw resume text into actionable insights and helps users improve job readiness through data-driven evaluations. It is scalable, user-friendly, and built on modern tools that allow future expansion into enterprise-scale analytics systems