

Resume compatibility checker

Name - Jatin Kumar Saxena

ID - 20244039

EMAIL - jatinkumar_20244039@students.iisertirupati.ac.in

Project Link - <https://github.com/Jatin-s16/Resume-check-portal-for-candidates>



Motivation for the work

While most resume screening tools are built for companies to shortlist candidates faster, very few focus on the candidates themselves. Job seekers are often left in the dark, not knowing how well their resume fits a job description or which version to use for which role. This project aims to flip that approach by giving candidates a transparent, NLP tool to assess their own resumes. By comparing multiple versions against specific job descriptions, they can understand compatibility, identify gaps, and improve their chances, all before they even hit apply.

Current Progress Made

I've made great progress on building the Resume Compatibility Checker over the last few months. The goal of this tool is to help job seekers check how well their resume matches a job description, something that's usually only available on the recruiter's side. Most platforms today focus on helping companies filter resumes efficiently, but very few offer candidates the ability to analyze and improve their resumes based on actual job requirements. To begin with, I developed a simple command-line interface where users can upload their resume and a job description in some specified format. As of the latest version, the tool lets users upload both their resume and a job description in PDF format through a **Streamlit-based interface**.

The platform performs text extraction using **PyMuPDF**, followed by a robust preprocessing pipeline - removing stopwords, punctuation, and performing lemmatization. This step ensures that even complex resume layouts are handled correctly and turned into plain text for analysis. I used **SpaCy** and **regex-based methods** to identify and isolate the key sections of a resume: Skills, Experience, and Education. These sections were stored in structured dictionaries for further analysis.

To measure how well the resume content matches the job description, I started by using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization combined with cosine similarity. This helped catch common keywords between the two documents, but it wasn't great at understanding context or the deeper meaning behind phrases.

In order to fix that, I upgraded the system to use sentence embeddings via **Sentence Transformers** (specifically all-MiniLM-L6-v2), which are BERT-based models. This significantly improved the accuracy of the compatibility scores by capturing contextual meaning, synonyms, and sentence-level similarity. For example, they can recognize that "Data Analytics" and "Data Analysis" are the same thing, or "created ML models" in a resume is similar to "developed predictive systems" in a job posting, even though the wording is different. I embedded both resumes and JDs section-wise and calculated

cosine similarity for each section separately—assigning custom weights to **Skills (40%)**, **Experience (30%)**, and **Education (20%)**. This refined approach yielded more intuitive and relevant results.

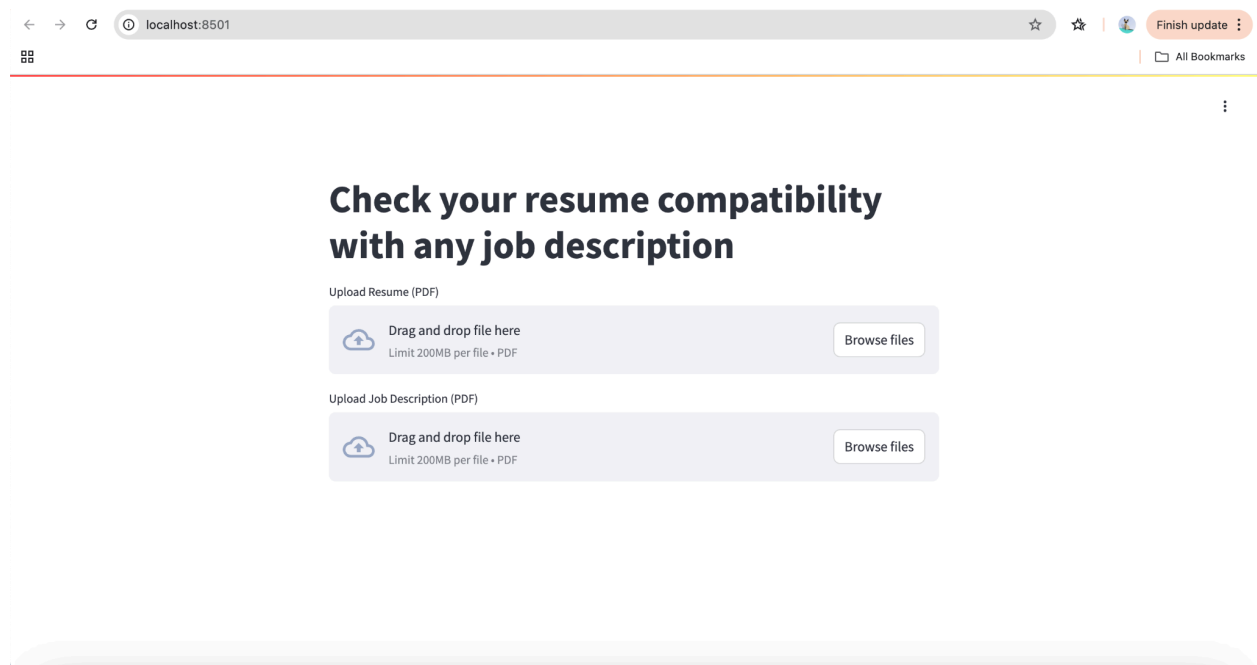
Here's a quick comparison of both approaches:

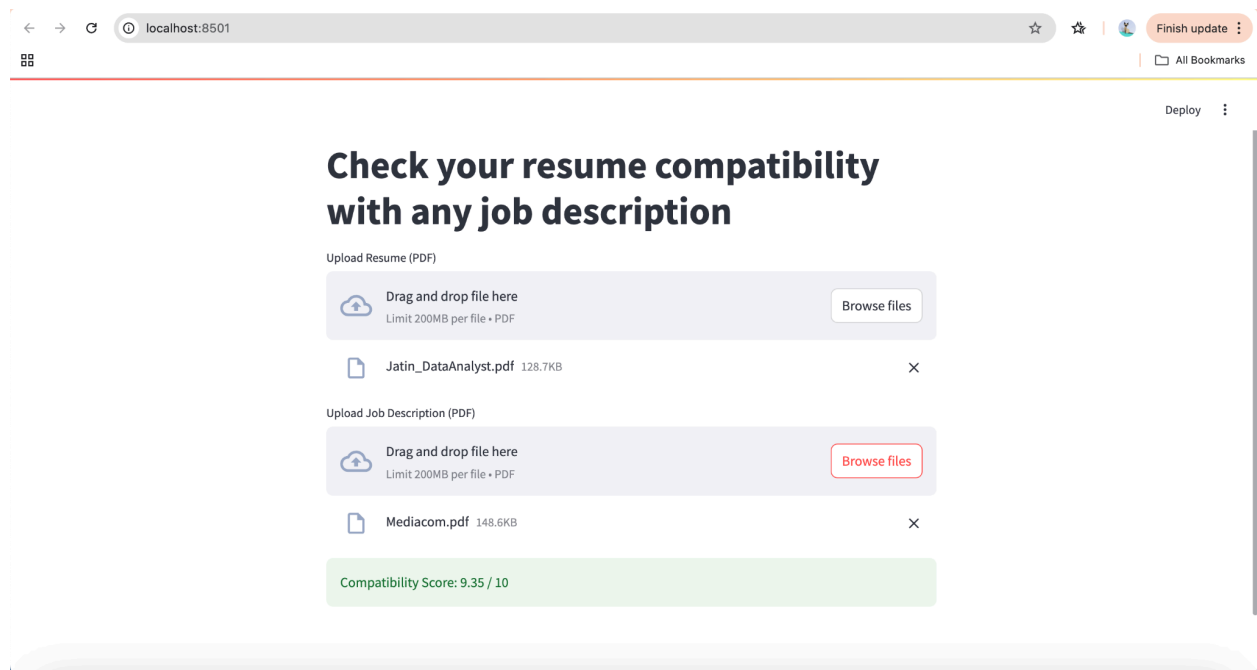
Technique Used	What it does	Result
TF-IDF + Cosine Similarity	Focuses on exact keywords	Basic scoring, misses context
Sentence Transformers	Looks at meaning and context	More accurate and useful results

The platform started as a command-line project for quick testing, but now it runs completely on Streamlit, which gives it a clean and interactive web interface. Users can easily upload files and get **real-time similarity scores out of 10**.

So overall, the processes taking place are -

- PDF file upload and text extraction for resumes and JDs.
- Skill, experience, and education section extraction from resumes.
- Sentence Transformer-based and cosine similarity scoring.
- Streamlit web app for user interaction.





Future Progress Possible till end-sem

There are several meaningful improvements planned for the next phase of the project, all aimed at making the platform more intelligent, insightful, and genuinely helpful to people. One of the biggest enhancements will be around how we interpret and weigh work experience. Right now, the platform does not differentiate between internships and full-time roles. However, in most real-world hiring situations, companies are very particular about the required number of years in full-time professional experience. So, I plan to build a filter that accurately identifies the full-time roles listed in a resume, estimates the total years of such experience, and compares it against the minimum experience criteria given in the job description. If the candidate falls short, the tool should not proceed with scoring, and instead inform the user that they're not eligible based on experience. This addition would make the platform far more realistic and aligned with actual hiring standards.

The scoring system will also be improved to dynamically assign weightage to different resume sections based on the candidate's experience level. For example, if the user is a fresher or recent graduate, more weight will be given to the kind of academic projects, certifications, and relevant coursework listed. On the other hand, for someone with industry experience, the focus should shift more toward their professional roles, responsibilities handled, and measurable outcomes achieved in previous jobs. This adaptive weighting mechanism will make the final compatibility score much more fair, nuanced, and informative.

Another feature in development is a feedback engine that provides users with detailed suggestions on what's missing or what can be improved in their resume to better match the uploaded job description. It will provide constructive insights like "You might want to highlight project management experience" or "Consider adding certain skills, which are frequently required for this role." This would help users not only assess their current fit but also actively work towards improving their application materials.

The platform's NLP backend will be upgraded. I'm planning to integrate Named Entity Recognition using advanced spaCy pipelines. This will allow better section detection (especially for loosely formatted resumes) and more accurate extraction of skills, organizations, job titles, and dates. This would help in validating timelines, differentiating training from full-time experience, and even spotting company names or tools frequently requested in job descriptions.

Finally, the deployment of the Streamlit app will be completed. While it currently works locally, the goal is to host it using Amazon Web Services so it becomes accessible to anyone with a browser.

Declaration

"I hereby declare that the idea for this project is my original idea. The report is also original, and I give my consent to check this report for similarity and plagiarism".