Project Title

Resume Ranker: Intelligent Resume Screening using NLP

Objective

The primary goal of this project is to automate the resume screening process by ranking resumes based on their relevance to a specific job description. This reduces manual effort for recruiters and helps shortlist the most suitable candidates quickly.

Problem Statement

Recruiters often receive hundreds of resumes for a single job opening. Manually reviewing and filtering these resumes is time-consuming and prone to human bias. There is a need for an AI-powered system that can automatically evaluate resumes and highlight the most relevant candidates.

Methodology

Step 1: Data Collection
- Input Resumes: Candidates' resumes in PDF/DOCX format.
- Job Description (JD): A text file containing the required skills, qualifications, and responsibilities.

Step 2: Text Extraction
- Extract raw text from resumes using libraries like:
- PyPDF2 / pdfminer (for PDF) - docx (for Word files)

Step 3: Preprocessing
- Clean and preprocess extracted text:
- Lowercasing
- Removing stop words, special characters, and numbers- Tokenization & Lemmatization

Step 4: Feature Engineering
- Convert text into numerical vectors using:
- TF-IDF (Term Frequency–Inverse Document Frequency) - OR Word Embeddings (spaCy / BERT embeddings)

Step 5: Similarity Scoring
- Compare each resume with the job description using:
- Cosine Similarity
- The higher the similarity score → the more relevant the resume.

Step 6: Resume Ranking
- Sort resumes based on similarity scores.
- Output: Ranked list of resumes from best match → least match.

System Architecture

User Input:
- Resumes (PDF/DOCX)
- Job Description (Text)

Processing Pipeline:
- Text Extraction → NLP Preprocessing → Vectorization → Similarity Calculation

Output:
- Ranked resumes with similarity percentage (e.g., Resume A: 87%, Resume B: 75%, etc.)

Tools & Technologies Used

- Programming Language: Python
- Libraries:
- PyPDF2, docx → Resume text extraction
- NLTK, spaCy → NLP preprocessing
- scikit-learn → TF-IDF, cosine similarity
- pandas → Data handling
- IDE: Jupyter Notebook / VS Code

Results

- The system successfully ranked resumes based on job description relevance.
- For example:
- Resume 1 → 89% match
- Resume 2 → 77% match
- Resume 3 → 65% match
- Recruiters can instantly shortlist the top 5–10 resumes instead of reading through hundredsmanually.

Applications

- HR & Recruitment automation
- University placement cell screening
- Freelance platforms (matching proposals with job postings) - Job portals (AI-based candidate ranking)

Future Enhancements

- Integration with ATS (Applicant Tracking Systems)
- Use of transformer models (BERT, RoBERTa) for better semantic matching
- Adding support for multilingual resumes
- Visual Dashboard (using Streamlit/Power BI) to display ranking results

Conclusion

The Resume Ranker project demonstrates how NLP and machine learning can be applied to automate resume screening. By ranking resumes against a job description, the system improves recruitment efficiency, reduces bias, and saves time for HR professionals.