Resume Ranking System Using Semantic Similarity

Author: JOSHI YASH AMISHKUMAR
Date: July 25, 2025

Contents

Abstract

Recruitment teams spend substantial time screening resumes for job openings. Manual resume sorting is time-consuming and subjective. This project presents a Resume Ranking System that automatically ranks uploaded resumes based on semantic similarity to a job description, using state-of-the-art NLP. The system streamlines HR workflows and aids in shortlisting relevant candidates efficiently.

1 Introduction

Organizations receive numerous resumes per job posting, which makes evaluating candidates a challenge. Traditional keyword-based filtering lacks the ability to understand context and meaning. This project leverages semantic similarity—comparing meaning, not just keywords— to automate and improve resume ranking.

2 Tools Used

- **Flask:** Web framework for the application interface.
- **pdfplumber, PyMuPDF:** Extract text from PDFs.
- **clean-text:** Text cleaning and standardization.
- **KeyBERT:** Extracts job-relevant keywords.
- **Sentence-Transformers:** For semantic embeddings and similarity.
- **scikit-learn, pandas:** Data handling and reporting.
- **HTML/CSS:** User interface design.

3 Steps Involved in Building the Project

1. **Web App Setup:** Built with Flask to enable resume uploads and job description input.
2. **Resume Upload & Extraction:** Users upload PDFs, from which text is extracted using pdfplumber or PyMuPDF.
3. **Text Preprocessing:** All texts are cleaned (standardized, noise removed).
4. **Keyword Extraction & Embedding:** KeyBERT finds job keywords; Sentence-Transformers generates semantic embeddings.
5. **Semantic Scoring & Ranking:** Cosine similarity between job and each resume produces a matching score; resumes ranked in descending order.
6. **Reporting:** A CSV report can be downloaded; results shown in a sortable HTML table.

4 Conclusion

The Resume Ranking System speeds up and smartens the hiring process by using semantic analysis to match resumes with job descriptions. The approach outperforms traditional keyword matching, is scalable, and can be extended to include more file formats and scoring methods.