CV Parser and Evaluator

Documentation

Introduction

The goal of this project is to create a Python-based application that parses and evaluates CVs against job descriptions using Natural Language Processing (NLP) techniques. The application will take a CV file and a job description file (Both files in PDF format) as input and output a score indicating how well the CV matches the job description.

Additionally, we may implement optional enhancements such as ranking multiple CVs, Structure folder, highlighting matching skills or experiences, and performing sentiment analysis.

Approach:

- Flowchart:

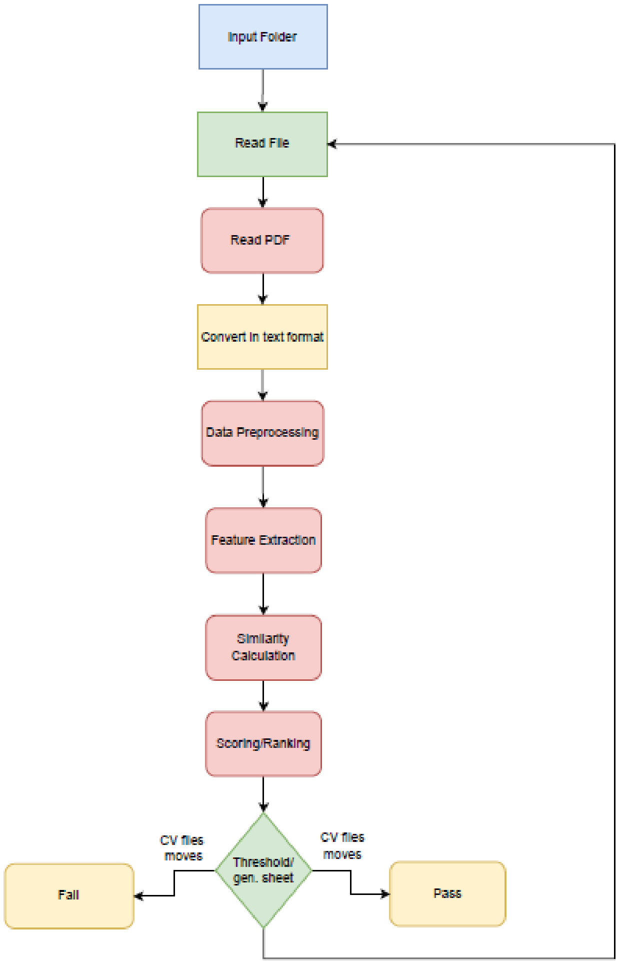


Fig 1 . Flowchart

- **DATA PREPROCESSING**

  Before applying any NLP techniques, we need to preprocess the CVs and job descriptions. Because we need to remove the unnecessary Stuff from data like Special characters, Links , supporting words etc.

  So we can reduce the noise in data.

  For Preprocessing, This includes the following steps:

  - **Tokenization**: Splitting text into individual words or tokens.
  - **Lowercasing**: Converting all text to lowercase for consistent matching.
  - **Stopword Removal**: Removing common words like "a," "the," "in," etc., that do not carry much meaning.
  - **Lemmatization**: Reducing words to their root forms to handle variations.

- **FEATURE EXTRACTION**

  To compare CVs and job descriptions, we need to convert text data into numerical features.

  Common methods include:

  - TF-IDF (Term Frequency-Inverse Document Frequency): Assigning weights to words based on their importance in the document convert in a vector. (Currently Using best method for comparsion)
  - Word Embeddings (e.g., Word2Vec, GloVe): Representing words as dense vectors.

  In current task, I am using the **TF-IDF VECTORIZER** method to extract the feature because future we are using in the cosine similarity method so it will be a good fit for it.

  Working of TF-IDF : Term Frequency – Inverse Document Frequency (TF-IDF).

  - ➢ It is a statistical method in NLP and information retrieval also It measure the importance of a term within a document.
  - ➢ In a simple manner it convert the words in numerical (vectorized form) which can used in ML/DL models.

- **Similarity Calculation**

  For the evaluation, need to calculate the similarity between the CV and job description using various distance or similarity metrics:

  - Cosine Similarity: Measures the cosine of the angle between two vectors.(Currently Using)
  - Euclidean Distance: Measures the straight-line distance between two points in vector space.

In current task, I am using using the Cosine Similarity method to find the similarity between CV and Job description. This method is widely use for Founding the similarity in NLP.

The combination of TF-IDF and Cosine Similarity is one of the best combination to use for similarity in the text part.

Working of Cosine Similarity:

- Cosine similarity measures the similarity between the vectors.
- It is measured by the cosine of the angle between vectors and determines whether vectors are pointing in roughly the same direction.
- It is often used to measure document similarity in text analysis.

The cosine Similarity mainly works with the vectors part that why I choose the TD-IDF feature extraction, method so both can support each other.

Cosine of the angle: Cosine Similarity calculates the cosine of the angle between two vectors. This angle is measured in the high-dimensional vector space and indicates how similar or dissimilar the two vectors are. The formula for cosine similarity between two vectors x and y is:

$$S(x, y) = x . y / ||x|| X ||y||$$

The cosine similarity is beneficial because even if the two similar data objects are far apart by the Euclidean distance because of the size, they could still have a smaller angle between them. Smaller the angle, higher the similarity.

- **SCORING**
  - The output score is calculated based on the similarity metric. A higher score indicates a better match between the CV and job description.
  - So the range of the relation metric is -1 to 1. To convert into percentage I apply small formula that converts into a readable Score.
  - Based On the readable score (%) I able to give the Rank to the CV.
  - Based on the Score(%) I set a threshold, so the resume is suitable of the job description or not (pass or fail).

OPTIONAL ENHANCEMENTS:

- Implement a ranking mechanism to rank multiple CV based on their scores(%) for given job description, so when it write the results in sheet I perform ‘reverse operation’ on score.
- For Additional insights, implement through Name entity recognition(NER) or Keyword extraction to identify to highlight specific skills or experience in the CV (initialize “spacy” with pre-trained model).
- Sentiment analysis to evaluate overall CV, implement through the ‘Textblob’ mainly use the sentiment analysis. This can provide additional insights into the candidate’s attitude for the job.

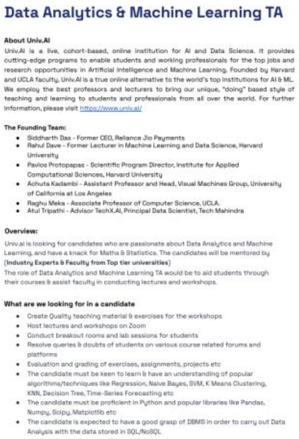
Folder Structure:

I have created a folder structure as :

- Input : Contains all Cv’s which need to evaluate.
- Job_desc :Contain the Job Description file.
- Output :The generated final is store in output folder and the CV’s
  - Pass : Contains all the pass CV’s.
  - Fail : Contains all the failed CV’s

Input File:

Job description input :



Data Analytics & Machine Learning TA

About Univ.AI

Univ.AI is a live, cohort-based, online institution for AI and Data Science. It provides cutting-edge programs to enable students and working professionals for the top jobs and research opportunities in Artificial Intelligence and Machine Learning. Founded by Harvard and UCLA faculty, Univ.AI is a true online alternative to the world’s top institutions for AI & ML. We employ the best professors and lecturers to bring our unique, “doing” based style of teaching and learning to students and professionals from all over the world. For further information, please visit https://www.univ.ai/

The Founding Team:

- Siddharth Das - Former CEO, Reliance Jio Payments
- Rahul Dave - Former Lecturer in Machine Learning and Data Science, Harvard University
- Pavlos Protopapas - Scientific Program Director, Institute for Applied Computational Sciences, Harvard University
- Achuta Kadambi - Assistant Professor and Head, Visual Machines Group, University of California at Los Angeles
- Raghu Meka - Associate Professor of Computer Science, UCLA.
- Atul Tripathi - Advisor TechX.AI, Principal Data Scientist, Tech Mahindra

Overview:

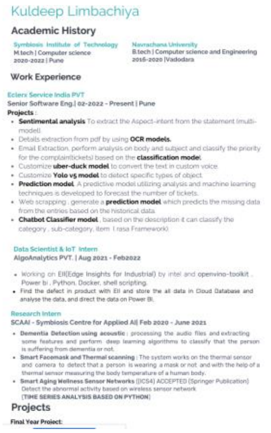
Univ.ai is looking for candidates who are passionate about Data Analytics and Machine Learning, and have a knack for Maths & Statistics. The candidates will be mentored by (**Industry Experts & Faculty from Top tier universities**)

The role of Data Analytics and Machine Learning TA would be to aid students through their courses & assist faculty in conducting lectures and workshops.

What are we looking for in a candidate

- Create Quality teaching material & exercises for the workshops
- Host lectures and workshops on Zoom
- Conduct breakout rooms and lab sessions for students
- Resolve queries & doubts of students on various course related forums and platforms
- Evaluation and grading of exercises, assignments, projects etc
- The candidate must be keen to learn & have an understanding of popular algorithms/techniques like Regression, Naive Bayes, SVM, K Means Clustering, KNN, Decision Tree, Time-Series Forecasting etc
- The candidate must be proficient in Python and popular libraries like Pandas, Numpy, Scipy, Matplotlib etc
- The candidate is expected to have a good grasp of DBMS in order to carry out Data Analysis with the data stored in SQL/NoSQL

CV input :



Kuldeep Limbachiya

Academic History

Symbiosis Institute of Technology
M.tech | Computer science
2020-2022 | Pune

Navrachana University
B.tech | Computer science and Engineering
2016-2020 |Vadodara

Work Experience

Eclerx Service India PVT
Senior Software Eng.| 02-2022 - Present | Pune
Projects :

- **Sentimental analysis** To extract the Aspect-intent from the statement (multi-model)
- Details extraction from pdf by using **OCR models.**
- Email Extraction, perform analysis on body and subject and classify the priority for the complain(tickets) based on the **classification model**.
- Customize **uber-duck model** to convert the text in custom voice.
- Customize **Yolo v5 model** to detect specific types of object.
- **Prediction model**. A predictive model utilizing analysis and machine learning techniques is developed to forecast the number of tickets.
- Web scrapping , generate a **prediction model** which predicts the missing data from the entries based on the historical data.
- **Chatbot Classifier model** , based on the description it can classify the category , sub-category, item (rasa Framework).

Data Scientist & IoT Intern
AlgoAnalytics PVT. | Aug 2021 - Feb2022

- Working on EII(Edge Insights for Industrial) by intel and openvino-toolkit , Power bi , Python, Docker, shell scripting.
- Find the defect in product with EII and store the all data in Cloud Database and analyse the data, and direct the data on Power BI.

Research Intern
SCAAI - Symbiosis Centre for Applied AI| Feb 2020 - June 2021

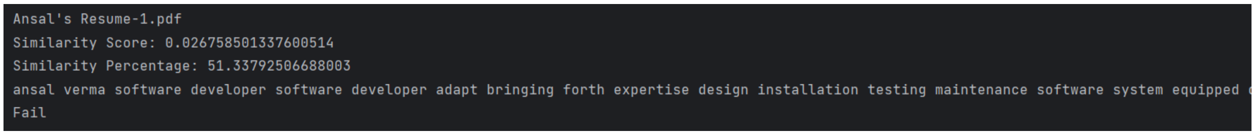
- **Dementia Detection using acoustic** : processing the audio files and extracting some features and perform deep learning algorithms to classify that the person is suffering from dementia or not.
- **Smart Facemask and Thermal scanning** : The system works on the thermal sensor and camera to detect that a person is wearing a mask or not and with the help of a thermal sensor measuring the body temperature of a human body.
- **Smart Aging Wellness Sensor Networks** ((ICS4) ACCEPTED (Springer Publication) Detect the abnormal activity based on wireless sensor network (**TIME SERIES ANALYSIS BASED ON PYTHON**)

Projects

Final Year Project:

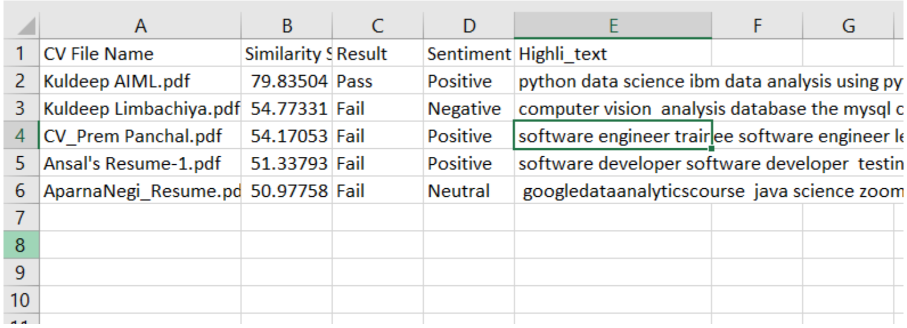
Output File :

Console output :



```
Ansal's Resume-1.pdf
Similarity Score: 0.026758501337600514
Similarity Percentage: 51.33792506688003
ansal verma software developer software developer adapt bringing forth expertise design installation testing maintenance software system equipped
Fail
```

Generate CSV:



	A	B	C	D	E	F	G
1	CV File Name	Similarity S	Result	Sentiment	Highli_text		
2	Kuldeep AIML.pdf	79.83504	Pass	Positive	python data science ibm data analysis using py		
3	Kuldeep Limbachiya.pdf	54.77331	Fail	Negative	computer vision analysis database the mysql c		
4	CV_Prem Panchal.pdf	54.17053	Fail	Positive	software engineer trainee software engineer le		
5	Ansal's Resume-1.pdf	51.33793	Fail	Positive	software developer software developer testin		
6	AparnaNegi_Resume.pd	50.97758	Fail	Neutral	googledataanalyticscourse java science zoom		
7							
8							
9							
10							

FUTURE IMPROVEMENTS:

In the future, we can explore the following improvements:

- **Semantic Analysis** : If Not getting good accuracy we can create some operation through regex/ML to extract the **Experience, Education, Skills** parts from the CV and Do the hole process.
- **Improvement of Accuracy** : If we find that this approach is not providing satisfactory results, we can explore more advanced models like BERT-based embeddings, Minkowski Distance, Hamming Distance, etc. Fine-tuning the model on domain-specific data for better matching.
- **Feedback Loop**: Implement a feedback mechanism that allows users to provide feedback on the accuracy of parsed information. Use this feedback to continually improve the parser. Incorporating user feedback and continuous learning to enhance matching accuracy.
- **Scalability -** We can leverage the power of multiprocessing using multi threads to handle larger chunks of incoming data with ease and improve the efficiency of the script Handling different file formats (e.g.,Doc , Image, txt) for CV parsing.
- **Error Handling**: Implement robust error handling to gracefully handle situations where the parser encounters unexpected or malformed CV.
- **Integration -** HR software can be integrated with this for seamless use in recruitment workflows

CONCLUSION:

This project aims to demonstrate the ability to develop a CV parser and evaluator using NLP techniques. While the initial implementation will be basic, it provides a foundation for further enhancements and improvements based on real-world requirements and user feedback.