JOB PORTAL USING NLP

Main Project Report

Submitted by

CHRISTEENA PAUL

FIT21MCA-2048

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications
Of
A P J Abdul Kalam Technological University

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY-683577, ERNAKULAM(DIST)

MAY 2023

DECLARATION

I, **CHRISTEENA PAUL**, hereby declare that the report of this project work, submitted to the Department of Computer Applications, Federal Institute of Science and Technology (**FISAT**), Angamaly in partial fulfillment of the award of the degree of Master of Computer Application is an authentic record of my original work.

The report has not been submitted for the award of any degree of this university or any other university.

Date : 29/04/2023

Place: Angamaly **Christeena Paul**

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)®

ANGAMALY, ERNAKULAM-683577

DEPARTMENT OF COMPUTER APPLICATIONS

FISAT

Focus on Excellence

CERTIFICATE

This is to certify that the project report titled **"JOB PORTAL USING NLP"** submitted by **CHRISTEENA PAUL [Reg No:FIT21MCA-2048]** towards partial fulfillment of the requirements for the award of the degree of Master of Computer Applications is a record of Bonafide work carried out by her during the year 2023.

Ms. Manju Joy	**Dr. Deepa Mary Mathews**
Project Guide	Head of the Department

Submitted for the viva-voice held on……………………….at……………………….

Examiner :

ACKNOWLEDGEMENT

Gratitude is a feeling which is more eloquent than words, more silent than silence. To complete this project work I needed the direction, assistance and co-operation of various individuals, which is received in abundance with the grace of God.

I hereby express my deep sense of gratitude to **Dr. Manoj George**, Principal, FISAT and **Dr. Mini P R**, Vice principal, FISAT, for allowing me to utilize all the facilities in the college. My sincere thanks to **Dr. Deepa Mary Mathews**, Head of the Department of Computer Applications, FISAT, who had been a source of inspiration. During the period of my project work, I have received generous help from **Ms.Manju Joy**, my Project Guide, **Dr. Santhosh Kottam**, my scrum master which I like to put on record here with deep gratitude and great pleasure.

Here I express my heartfelt thanks to all the faculty members in my department for their constant encouragement and never-ending support throughout the project. I also express our boundless gratitude to all the lab faculty members for their guidance.

Finally, I wish to express a whole heart-ed thanks to my parents, friends and well-wishers who extended their help in one way or other in preparation of my project. Besides all, I thank God for everything.

ABSTRACT

In this competitive era, getting right education and right job is always a challenge. Organization who are in need of skilled people in certain departments also find it difficult to identify right candidate with good talented skill set. In the proposed system, I propose 2 logins namely job seeker and employer. The proposed system tries to make the recruitment process simpler and efficient by integrating text mining and natural language processing techniques. The proposed architecture consists of unique and essential features like study materials, de-duplication process, resume analysis and weightage analysis. The employer can upload study materials while posting the job requirement so that the job seeker will have a fair knowledge of the exact job role. The recommendation of exact profile based on the skill required is processed using collaborative filtering algorithm. To optimize the cloud storage we have integrated de-duplication technique to eliminate saving same resume n number of times which would increase encryption cost and storage. The de-duplicaton process is performed using Proactive Replica Checking approach (PRCR). Also applying natural processing techniques in both job seeker and employer side provides efficient results saving much of time. In the employer side, I use web crawler to extract job description and requirements. In the job seeker side, once the resume is posted, stop word filtering and text segmentation is performed. After text segmentation, the scoring is provide based on the education, work experience, skills, personality traits and frequency of degree. Finally my proposed system provides a recommendation system for the upcoming generation in which degree of education major job requirements are coming.

CONTENTS

Chapter I
INTRODUCTION

1.1 SCOPE OF THE WORK

In this competitive era, getting right education and right job is always a challenge. Organization who are in need of skilled people in certain departments also find it difficult to identify right candidate with good talented skill set. The proposed system tries to make the recruitment process simpler and efficient by integrating text mining and natural language processing techniques. The proposed architecture consists of unique and essential features like study materials, de-duplication process, resume analysis and weightage analysis. The employer can upload study materials while posting the job requirement so that the job seeker will have a fair knowledge of the exact job role. The recommendation of exact profile based on the skill required is processed using collaborative filtering algorithm. To optimize the cloud storage I have integrated de- duplication technique to eliminate saving same resume n number of times which would increase encryption cost and storage. The de-duplicaton process is performed using Proactive Replica Checking approach (PRCR). Also applying natural processing techniques in both job seeker and employer side provides efficient results saving much of time. In the employer side,we use web crawler to extract job description and requirements. In the job seeker side, once the resume is posted, stop word filtering and text segmentation is performed. After text segmentation, the scoring is provide based on the education, work experience, skills, personality traits and frequency of degree. Finally our proposed system provides a recommendation system for the upcoming generation in which degree of education major job requirements are coming.

Indian I.T sector is second largest candidate recruiting sector of our country. It contribute about 7.5% to our Gross Domestic Product(G.D.P) Our Proposed system is initiallyconcerned with the I.T sector of our country. It is mainly going to deal the Indian I.T industrybut if you talk about the pro version of our system it can be extended to various other commercial sector where, intake and elimination are in bulk like for Govermental Jobs.

Chapter II

PROOF OF CONCEPT

2.1 INTRODUCTION

Natural Language Processing (NLP) is an hot research area which is tremendously getting increased. NLP can be adopted in several area's which can help human is decision making and automation. NLP is applied in online product companies to mine the n number of reviews and make the customer decision making easier. NLP is applied to mine speech input to analyse the parameter and identify the meaning automatically. NLP has huge scope in day to day life especially to help interviewers. Implementation of NLP for interviewer, resume analysis is more advantageous. This is because it eliminates the interviewer don't treat the candidates with personal, mental, physical traits, external conditions, influence, referral etc. The efforts taken by the organization human resource department is also increasing in calling list of candidates, arrange infrastructure for interview venues and schedule interviewers for the interview process. Thus fulfilling candidate requirement might take much time and increased cost. Existing traditional approach leads the organization to innovate handling the process making work less complex. The innovation might include domains like text mining, natural language processing to make the process simple and efficient. Productive and good candidates are eliminated in the interview process by the interviewer due to their wring decisions, wrong prediction of candidate subjective views, personal emotion. Thus because of these parameters right candidate may be missed to get the right opportunity. Always hiring right candidate is an important factor for the success of the organization to obtain their mission and vision. Identifying right productive candidate is challenging for every human resource departments. Thus automating the interview process with NLP can reduce much time and make the decision making easier. Application of text mining technique to the candidate resume and projecting the list of candidate resume along with weightage of skills would help the human resource department in identifying the right candidate at short period of time.

2.2 REVIEW OF LITERATURES

Here is a more detailed review of literature on methods, results, accuracy, and comparison of the topic identification of job portal using natural language processing:

"A novel job portal with resume evaluation system based on text mining and NLP" by S. Ganesh Kumar, Ritvik Jain, Ashish Singh Chauhan. (2020)

Method: Natural Language Processing techniques have been used in implementing this approach – Named Entity Recognition (NER), Word embedding model, and Cosine similarity using which a resume and job will be matched. The NER model is used to extract useful entities from documents, which is enhanced by the word2vec model by making the system more generic and the similarity is calculated using the cosine similarity algorithm..

" **Differential Hiring using a Combination of NER and Word Embedding" by Suhas H E, Manjunath A E.(2020)**

Method: In the resumes undergo a series of analysis steps such as syntactic analysis, semantic analysis, and Lexical analysis. In each step of the analysis, attributes are extracted from which inferences are drawn, or the data extracted is cleaned to make it less noisy. The crux of the algorithm lies in the number of matched items from the existing candidate skill set database, which is continuously updated.

"Machine Learning approach for automation Resume Recommendation System" by Bhushan Kinge , Shrinivas Mandhare, Pranali Chavan, S. M. Chaware (2020)

Method: Pradeep Kumar Roy in their research ,created a system where they can minimize the cost of hiring new candidates for the job positions in the company. They focused on 3major problems in this process

- Picking the right candidates from the applicants
- Making sense of their CV’s
- Finding out if the candidate is fit for the job role.

They performed NER, NLP, and text classification using n-grams and used Machine Learning to perform the classification using the algorithms of Random Forest with 38.9% accuracy, Multinomial Naïve Bayes with 44.39%, Logistic Regression with 62.4%, and the highest accuracy was obtained by Linear Support Vector Machine Classifier with an accuracy of 78.53%

"Skill Finder: Automated Job-Resume Matching System" by Thimma Reddy Kalva. (2022)

Method: In research conducted by Thimma Reddy Kalva ,they have developed a custom dataset of 3000 jobs and 80 resumes from the website indeed using the web service API . This data is then used to rank the student's resumes comparing their skills required in the job, this is done using the Named Entity Recognition(NER) like Apache Open NLP and Stanford Name Entity Recognizer .The Skill finder efficiently matches the Resumes according to the job role posted and successfully sends emails to the desired candidates

"ResumeNET : A Learning-based Framework for Automatic Resume Quality Assessment" by Yong Luo . (2019)

Method: In research conducted by Yong Luo [3], they have developed a custom dataset of 10,343 resumes which was acquired by a private resume management company. 98.82% (i.e 10,221 resumes) data is unlabeled and the remaining 1.18% (i.e. 122 resumes)data is labeled in 2 categories positive and negative,33 and 89 of them are labeled as positive and negative.

2.3 LIMITATION OF EXISTING MODELS/ SYSTEMS

The problem is that the present are not much flexible and efficient and time saving. It requires candidate, to fill the forms online than also you might not get the genuine information of the candidate. Beside Where our system which saves the time of the candidate by providing to upload there resume in any format preferable to the candidate beside all the information in the resume our system will detect all its activity from the candidate social profile which will give the best candidate for that particular job and candidate will also be satisfied because he will get job in that company which really appreciates candidates skill and ability. On the other hand we are providing same kind of flexibility to the client company.

Chapter III

SYSTEM ANALYSIS AND DESIGN

3.1 SYSTEM ANALYSIS

3.1.1 INTRODUCTION

Systems analysis is "the process of studying a procedure or business to identify its goal and purposes and create systems and procedures that will efficiently achieve them". Another view sees system analysis as a problem-solving technique that breaks down a system into its component pieces, and how well those parts work and interact to accomplish their purpose.

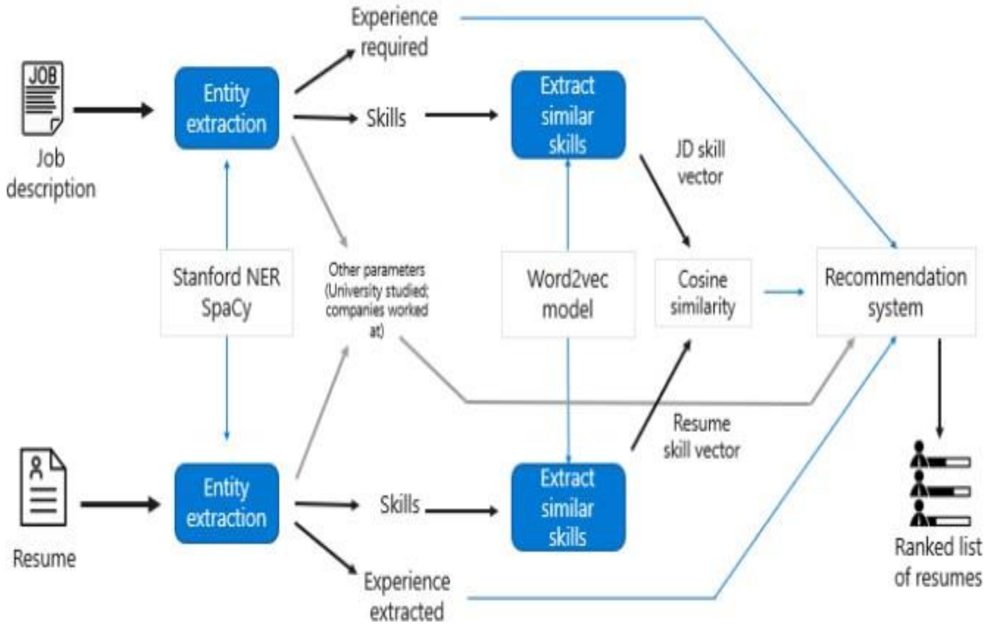
System analysis is used in every field where something is developed. Analysis can also be a series of components that perform organic functions together, such as systems engineering. Systems engineering is an interdisciplinary field of engineering that focuses on how complex engineering projects should be designed and managed.

3.1.2 PROPOSED SYSTEM OBJECTIVES

The overall flow of the process is depicted as in below. It depicts how the end to end process occurs given a Job opening description and a resume. It has 3 essential steps Named Entity Recognition, Word2vec, and Cosine similarity.

Named Entity Recognition

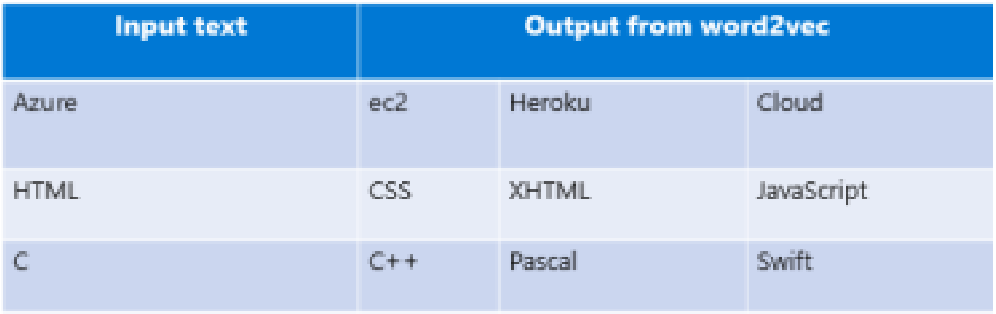
As explained above, a NER model is trained, and the model is prepared. The NER model takes to input the text of the document, which can be a JD/Resume text and recognizes entities in them.



Flow the process

Word2vec model

The skills extracted from the NER are provided as input to the word2vec model yields similar skills to the skills extracted by the NER. The most important part played by the word2vec is that it gives weightage to a resume even if that candidate has similar skills to the ones mentioned in the job opening description.



Input text	Output from word2vec		
Azure	ec2	Heroku	Cloud
HTML	CSS	XHTML	JavaScript
C	C++	Pascal	Swift

Outputs of the trained word2vec model

As depicted in the above table, if a job opening has a skill required such as Azure, when this word is given as input to the word2vec model words such as ec2, Heroku, and cloud have been obtained as output. In reality, each of these output words is accompanied by a floating number in the range of 0 to 1, indicating the closeness to the input word, which plays an essential part in the next step of the process.

Cosine Similarity

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

Cosine similarity expression

Cosine similarity is the final step of the process, which determines how much a given resume and job opening matches. Cosine similarity literally translates to the angle between the 2 vectors representing a JD (Job Description) and a candidate's resume. Based on the research done in , the best way to apply cosine similarity between the resume and JD was ideated and summarized below. The steps followed to obtain a list of ranked candidates is as follows:

1. The skills extracted from each of the JD and resume were obtained along with similar skills obtained from the word embedding model.

2. The set of union of all the skills was taken, and the vector length was set to the length of this set.

3. 2 vectors are created for a JD and resume, respectively. For each skill in the union of skills, the score of each skill is taken as 1 if it is present directly in the document. Otherwise, the score is taken as per the output of the word2vec model.

4. Using the above formula, as in figure 6, the cosine similarity is calculated against each JD and sorted according to the score. The top 5 to 10 candidates are recommended to the recruiter.

3.1.3 HARDWARE AND SOFTWARE REQUIREMENTS

The selection of software is very important in the existence and proper working of any software. When selecting software, the size and capacity requirements are also important. Below are some of the software that is required for the system.

platform	Anaconda
Programming languages	Python
OS	Windows 11

The most common set of requirements defined by any operating system or software application is the physical computer re- sources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. I require much different software to make the application which is in making to work efficiently. It is very important to select the appropriate software so that the software works properly.

Processor	i5 or i7
RAM	8 GB or above
Hard Disk	512
Mouse	3D optical mouse
Keyboard	Standard 108 keys

3.2 SYSTEM DESIGN

3.2.1 INTRODUCTION

System design is the process of defining the architecture, interfaces, and data for a system that satisfies specific requirements. System design meets the needs of your business or organization through coherent and efficient systems. Once your business or organization determines its requirements, you can begin to build them into a physical system design that addresses the needs of your customers. The way you design your system will depend on whether you want to go for custom development, commercial solutions, or a combination of the two.

System design requires a systematic approach to building and engineering systems. A good system design requires you to think about everything in an infrastructure, from the hardware and software, all the way down to the data and how it’s stored.

3.2.2 MODULE DESCRIPTION

This project includes mainly 4 modules. They are,

1. 1.Job seeker registration and profile management

This module allows job seekers to create a profile on the job portal and manage their personal and professional information, including their education, work experience, and skills.

2.Employer registration and job posting

This module enables employers to create an account on the job portal and post job openings. Employers can specify the job title, job description, required qualifications, and other details about the job.

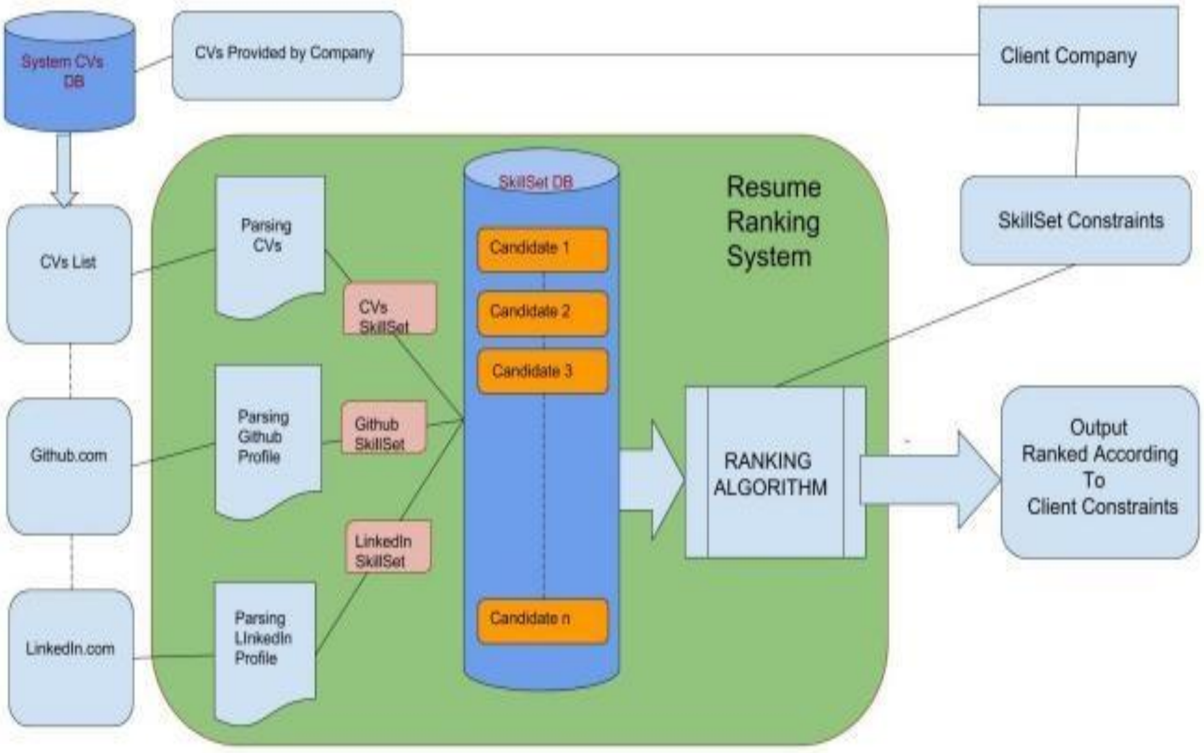
3.Job search and matching

This module allows job seekers to search for job openings based on keywords, location, industry, or other criteria. The module also includes a matching algorithm that recommends jobs to job seekers based on their skills and experience.

4.Analytics and reporting

This module provides analytics and reporting features for job seekers and employers. Job seekers can track their job search progress, including the number of applications submitted and the number of interviews scheduled. Employers can track the performance of their job postings and the effectiveness of their recruiting efforts.

3.2.3 SYSTEM ARCHITECTURE



The accuracy of the NER model depends on the data that is used to train the model and the pre-processing steps that have been applied to reduce the noise in the data and the accuracy of annotations in the labeled data. There are several steps which were followed to train a NER which should recognize technical skills from the given input text:

1. A collection of documents which is related to the problem at the handset of Job openings, set of resumes with the approval of candidates were taken.

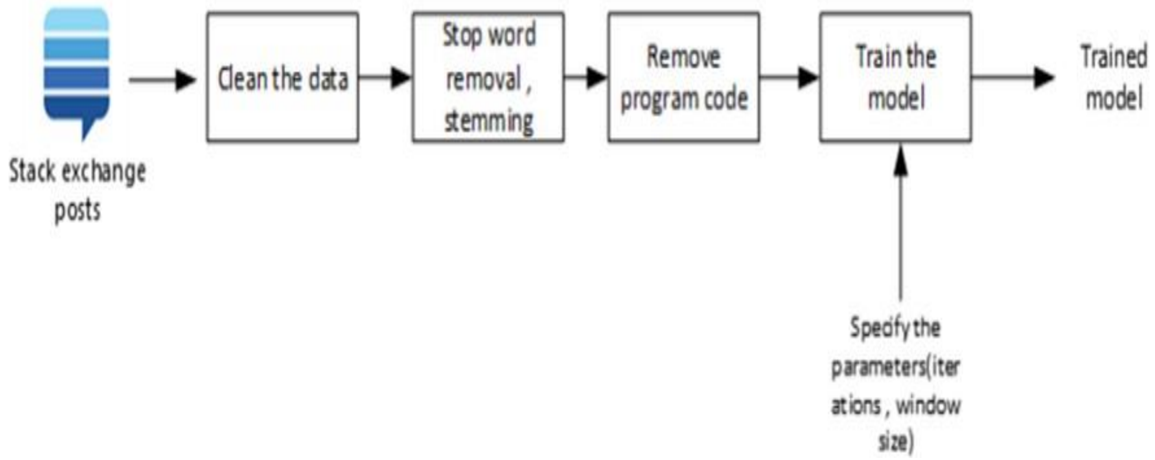
2. Series of pre-processing steps were applied to these documents such as stop word removal, stemming, punctuation removal, which yielded only words that are crucial to NER training.

3. A data dump of technical skills [used to tag the technical skills in each of these input documents.

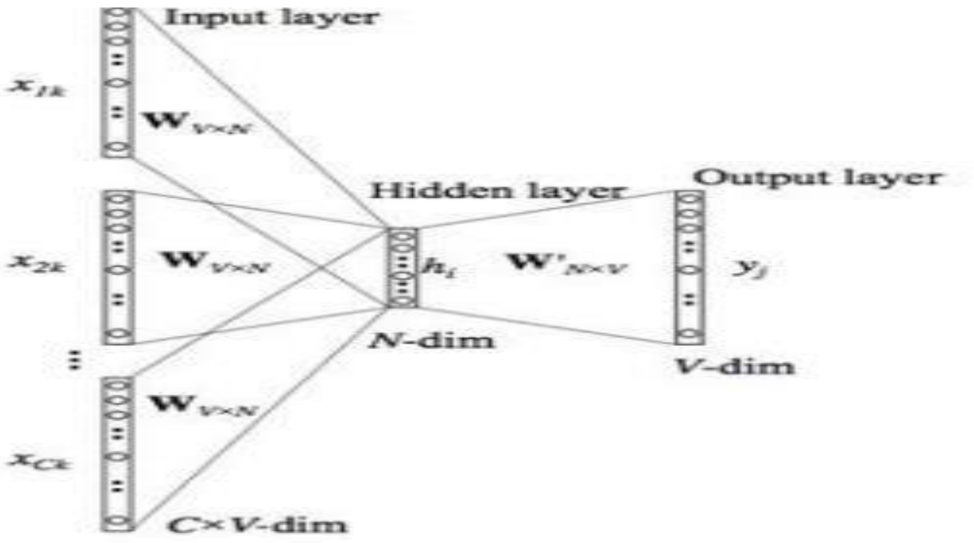
4. A Tab-separated value (TSV) file is generated for each of these annotated documents and provided as input to the Stanford NER package

5. After 20 iterations, the accuracy of the NER is optimized, and the model is obtained.

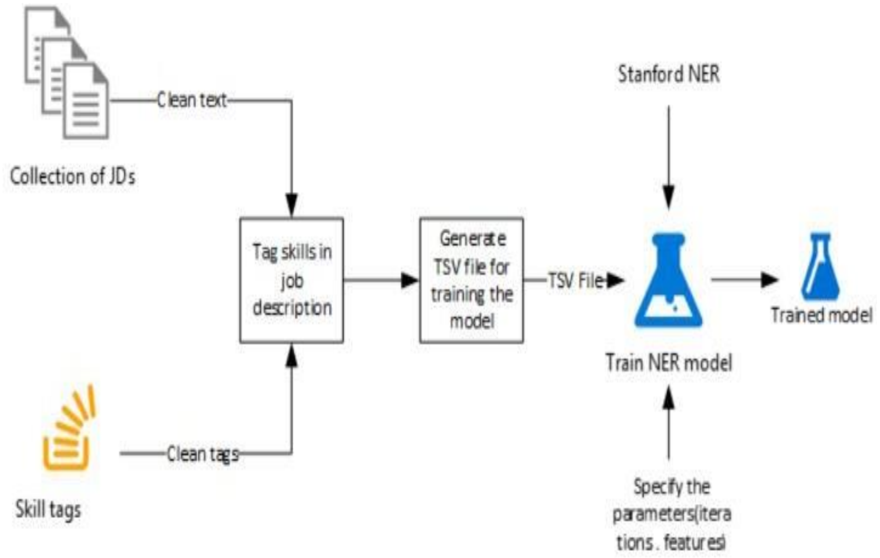
A conditional random field (CRF) based network provided by the Stanford NER package has been used to build the custom entity recognizer. As explained in , a Gibbs sampling method is used, which forms long-distance relationships with distant words, which enhances the performance of the model.



Training a Word2vec model



Word2Vec architecture

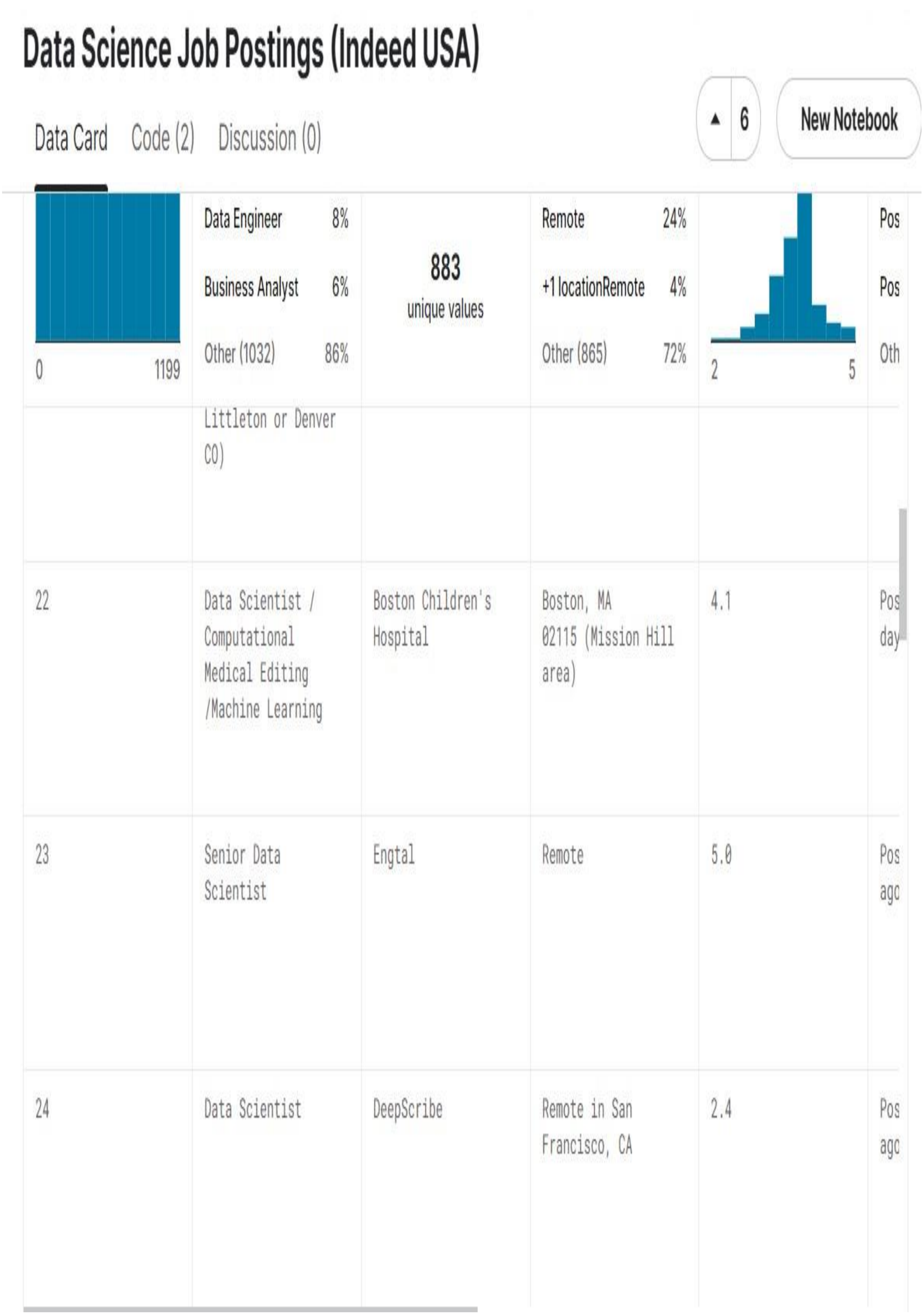

Collection of JDs
Clean text
Skill tags
Clean tags
Tag skills in job description
Generate TSV file for training the model
TSV File
Stanford NER
Train NER model
Trained model
Specify the parameters(itera tions. features)

Training a custom NER model

3.2.4 DATABASE/ DATASETS

The dataset contains job listings across many data science positions which includes data scientist, machine learning engineer, data engineer, business analyst, data science manager, database administrator, business intelligence developer and director of data science in the US. There are 1200 rows and 9 columns. The column headings are job title, company, location, rating, date, salary, description (summary), links and descriptions (full). The data was web scraped from indeed web portal on Nov 20, 2022 using the indeed API.

Datasets like this could help sharpen your skills in data cleaning, EDA, feature engineering, classification, clustering, text processing, NLP etc. There are many NaN entries in the salary column as most job listings do not provide salary info, can you come up with a way to fill those entries? The last column (descriptions) contains the full job description, with this at your disposal, there is an infinite number of features you could extract such as skill requirement, education, experience, etc.



Data Science Job Postings (Indeed USA)

Data Card Code (2) Discussion (0)

	Data Engineer 8% Business Analyst 6% Other (1032) 86%	883 unique values	Remote 24% +1 locationRemote 4% Other (865) 72%		Pos Pos Oth
	Littleton or Denver CO)				
22	Data Scientist / Computational Medical Editing /Machine Learning	Boston Children's Hospital	Boston, MA 02115 (Mission Hill area)	4.1	Pos day
23	Senior Data Scientist	Engtal	Remote	5.0	Pos ago
24	Data Scientist	DeepScribe	Remote in San Francisco, CA	2.4	Pos ago

3.3 RESULTS AND DISCUSSIONS

3.3.1 INTRODUCTION

Job Portal trait makes it easier for job seekers to decide on a specific career. Equally, online job search sites can help employers pace up their hiring process. It would be easy for recruiters or hiring managers to verify job applications online than physically scan printed resumes.

An Online Job Portal is an application where the job seekers can register themselves at the website and search jobs which are suitable for them where as the employers register with the website and put up jobs which are vacant at their company.

A job board enables employers to publish job offers for a position to be filled. For their search, jobseekers navigate through different websites in order to find a job.

3.3.2 TEST CASES

Some of the test cases that performed on the model to ensure its ability are as follows;

1. **test_case_1**: One of the top use cases of natural language processing is translation. The first NLP-based translation machine was presented in the 1950s by Georgetown and IBM, which was able to automatically translate 60 Russian sentences to English. Today, translation applications leverage NLP and machine learning to understand and produce an accurate translation of global languages in both text and voice formats.

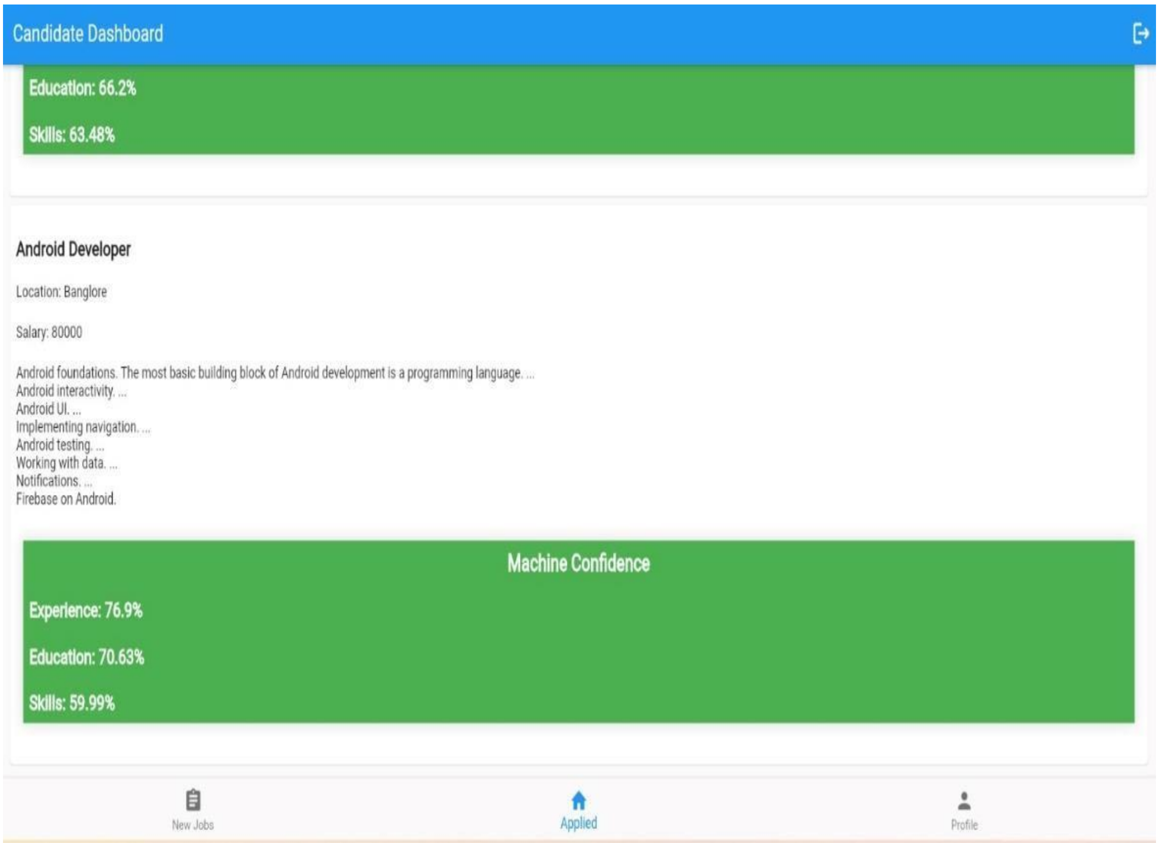
2. **test_case_2** : NLP is used to identify a misspelled word by cross-matching it to a set of relevant words in the language dictionary used as a training set. The misspelled word is then fed to a machine learning algorithm that calculates the word’s deviation from the correct one in the training set. It then adds, removes, or replaces letters from the word, and matches it to a word candidate which fits the overall meaning of a sentence.

3 **test_case_3**: Autocomplete, or sentence completion, combines NLP with certain machine learning algorithms (e.g. Supervised learning, Recurrent neural networks (RNN), or Latent semantic analysis (LSA)) in order to predict the likelihood of using a following word or sentence to complete the meaning.

4 **test_case_4**: Conversational AI is the technology that enables automatic conversation between computers and humans. It is the heart of chatbots and virtual assistants like Sirior Alexa. Conversational AI applications rely on NLP and intent recognition to understand user queries, dig in their training data, and generate a relevant response.

5 **test_case_5**: Chatbots have numerous applications in different industries as they facilitate conversations with customers and automate various rule-based tasks, such as answering FAQs or making hotel reservations. Haptik, a provider of conversational AI services, works with a number of Fortune 500 companies, including Disney, Tata, HP, Unilever, Zurich, and others. Haptik's chatbots and intelligent virtual assistants (IVAs) assist its clients' businesses in boosting profits and user engagement whilecutting costs.

3.3.3 SCREEN SHOTS OF IMPORTANT RESULTS



Education: 66.2%

Skills: 63.48%

Android Developer

Location: Banglore

Salary: 80000

Android foundations. The most basic building block of Android development is a programming language. ...
Android interactivity. ...
Android UI. ...
Implementing navigation. ...
Android testing. ...
Working with data. ...
Notifications. ...
Firebase on Android.

Machine Confidence

Experience: 76.9%

Education: 70.63%

Skills: 59.99%

3.3.4 RESULTS

NER models are usually measured for their accuracy by comparing the outputs of the NER model and human labeled counterparts. Its particularly difficult to measure the accuracy in this case because we are identifying words, so what happens in the case the output words overlap with the correct output either partially or subsumes it. Exact-match Evaluation [16] mainly involves two subtasks: boundary detection and type identification. More specifically, the number of False positives (FP), False negatives (FN), and True positives (TP) are used to compute Recall, Precision, and F-score.

• False Positive (FP): an entity that is the output of the NER model but does not appear in the ground truth.

• False Negative (FN): an entity that is the output of the NER model but appears in the ground truth.

• True Positive (TP): an entity that is returned by a NER system and appears in the ground truth.

Precision indicates the percentage of our results, which are correctly recognized. Recall indicates the percentage of total entities correctly recognized by the NER model, as shown below.

$$\text{Precision} = \frac{\#TP}{\#(TP + FP)} \qquad \text{Recall} = \frac{\#TP}{\#(TP + FN)}$$

F-score takes both Precision and Recall and computes the harmonic mean of the two. It provides a more realistic measure of the performance of the model.

$$\text{F-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The optimization of models went through 4 main iterations which are explained in detail below:

- Iteration 1 Trained the NER with minimal preprocessing steps such as removal and punctuation initially with around 100 manually annotated resumes.

- Iteration 2 Text preprocessing supplemented with techniques like stop word removal and Stemming.

- Iteration 3 The dataset was increased from 100 to 200 manually annotated resumes increasing the training dataset.

- Iteration 4 Several optimizations are done. A dictionary of skills was maintained to consider skills with symbols in them, such as C#; the window size was increased to take into consideration more words on either side of the current word during the training of the NER.

The optimization of models went through 4 main iterations

which are explained in detail below:

- Iteration 1

Trained the NER with minimal preprocessing steps such as removal and punctuation initially with around 100 manually annotated resumes.

- Iteration 2

Text preprocessing supplemented with techniques like stop word removal and Stemming.

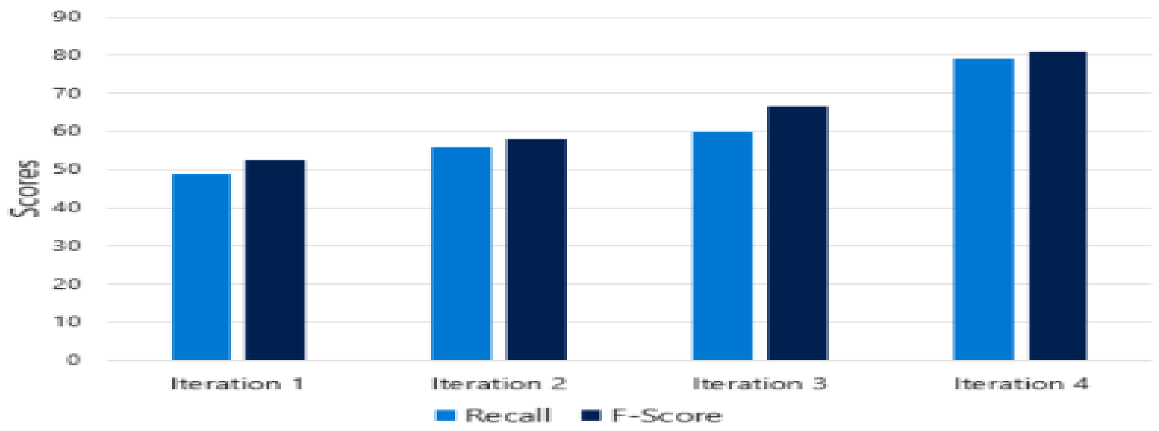
- Iteration 3

The dataset was increased from 100 to 200 manually annotated resumes increasing the training dataset.

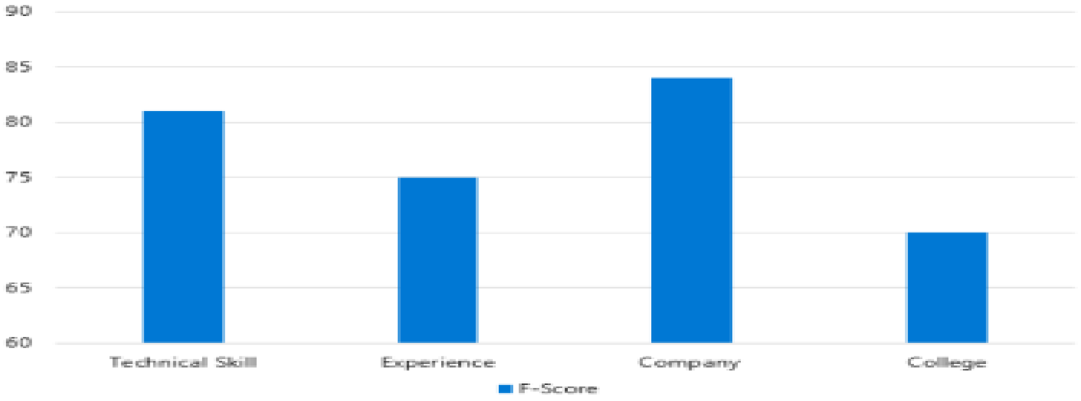
- Iteration 4

Several optimizations are done. A dictionary of skills was maintained to consider skills with symbols in them, such as

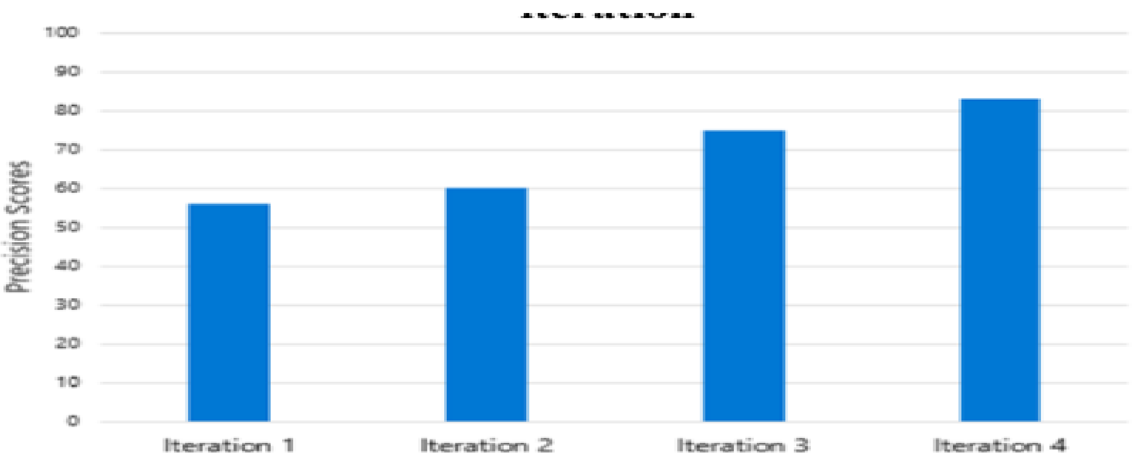
C#; the window size was increased to take into consideration more words on either side of the current word during the training of the NER.



Variation of Recall and F-score with each iteration



Variation of Recall and F-score with each iteration



Variation of Precision scores with each iteration

Chapter IV

SUMMARY

4.1 CONCLUSION

This project presents a method to match talent using Natural Language Processing techniques such as NER and word embedding. The system built on the proposed method is successfully abele to match jobs and candidates in the 2 scenarios discussed.

The novel part of the method being extraction of skills that are similar to the ones present in the document with the help of a word embedding model which makes the model more inclusive. Based on the metrics of the model performance, it can be said that technical skills are extracted with high enough accuracy. Also, the system is built in such a way that there is an absence of absolutely any kind of bias in the model and treat all the applicants relatively . Thus, this method of talent matching greatly enhances the work of a recruiter as well as the job application process, which alleviates the problem of going through thousands of job openings, making it a seamless process.

My project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted . This package shall prove to be a powerful package in satisfying all the requirement of the school . The objective of software planning is to provide a frame work that enables the manger to make reasonable estimate made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points

- A description of the background and context of the project and its relation to work already done in the area.
- Made statement of the aims and objective of the project.
- The description of purpose. Scope, and applicability.
- I define the problem o which we are working in the project.

- I describe the requirement specification of the system and the action that can be done on these things.
- I understand the problem domain and produce a model of the system, which describes operation that can be performed on thee system.
- I include features and operation in details. Including screen Layouts.
- I design user interface and security issues related to system.
- Finally the system is implemented and tested according to test cases.

4.2 FUTURE ENHANCEMENTS

Future work can be directed towards developing techniques of a feedback loop to increase the performance of the model continually. Feedback buttons are provided at the user interface level of the application where the recruiter can provide the feedback, and accordingly, the parameters of NER and word2vec training are tuned to obtain the maximum match between resume and job opening

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding.

- I can add printer in future.
- I can give more advance software for online job portal including more
- I will host the platform on online servers to make it accessible worldwide.
- Integrate multiple load balancers to distribute the loads of the system..
- Create the master and slave database .structure to reduce the overload of the database queries.
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.

The above mentioned points are the enhancement which can be done to increase the applicability and usage of this project. Here we can maintain the records of job and vacancy. Also, as it can be seen that now a days the players are versatile, that is so there is a scope for introducing a method to maintain the online job portal enhancement can be done to maintain all the job vacancy, resume , job seeker , interview.

I have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them . In the last we would like thanks all the person involved in the development of the system directly or indirectly. I hope that the project will serve its purpose for which it is develop there by underlying success of the process.

SAMPLE CODE

Main.py

```
import random
import spacy
import os
import gensim
from flask import Flask, request, jsonify, Response
from flask_mysqldb import MySQL
from flask_cors import CORS
import MySQLdb.cursors
from werkzeug.serving import WSGIRequestHandler
from werkzeug.utils import secure_filename
from PyPDF2 import PdfReader
app = Flask(_name_, static_folder="static")
WSGIRequestHandler.protocol_version = "HTTP/1.1"

nlp_ner = spacy.load("model-best")
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'job'
app.secret_key = 'job'

CORS(app)
mysql = MySQL(app)
# Load pre-trained word embedding model
model = gensim.models.keyedvectors.KeyedVectors.load("model\glove-gigaword.model")
```

```
def get_List(list,list2,type):
    job = []
    candidate = []
    print(list)
    print(list2)
    try:
        for i in list:
             job.append(i[type])
        for j in list2:
            candidate.append(j[type])
        similarity = model.n_similarity(job, candidate)
        return similarity
    except:
        return random.uniform(0.5, 0.85)



def score(email,job_id):
    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
    cursor.execute("SELECT * FROM candidate_edu where email='{}'".format(email))
    edu = cursor.fetchall()
    cursor.execute("SELECT skill FROM candidate_skills where
email='{}'".format(email))
    skill = cursor.fetchall()
       cursor.execute("SELECT * FROM candidate_exp where email='{}'".format(email))
    exp = cursor.fetchall()
    cursor.execute("SELECT * FROM job_edu where job_id={}".format(job_id))
    jedu = cursor.fetchall()
    cursor.execute("SELECT skill FROM job_skills where job_id={}".format(job_id))
    jskill = cursor.fetchall()
    cursor.execute("SELECT * FROM job_exp where job_id={}".format(job_id))
    jexp = cursor.fetchall()

    skill_score = get_List(jskill, skill, 'skill')

    print(skill_score)

    edu_score = get_List(jedu, edu, 'edu')

    print(edu_score)

    exp_score = get_List(jexp, exp, 'exp')

    print(exp_score)

    return
[round(exp_score*100,2),round(edu_score*100,2),round(skill_score*100,2)]


@app.route('/api/viewJobs/', methods=["POST"])

def view_apply():

    if request.method == "POST" and 'email' in request.form:

        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

        email = request.form['email']

        cursor.execute("select * from job_post,job_apply where job_apply.email = '" + email
+ "' and job_post.job_id ="

"job_apply.job_id")

        data = cursor.fetchall()

        return jsonify(data)

    else:

        return jsonify(status="Request error")

@app.route('/api/viewCandidates/', methods=["POST"])

def view_candidate():

    if request.method == "POST" and 'id' in request.form:

        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

        _id = request.form['id']

        cursor.execute("select * from job_post,job_apply,candidate where
job_apply.job_id={} and"

                        " candidate.email = job_apply.email

and job_post.job_id = job_apply.job_id".format(_id))
```

```
            data = cursor.fetchall()
            return jsonify(data)
        else:
            return jsonify(status="Request error")
@app.route('/api/applyJob/', methods=['POST'])
def apply_job():
        if request.method == "POST" and 'email' in request.form:
            email = request.form['email']
            edu  =  request.form['edu']
            exp  =  request.form['exp']
            skill = request.form['skill']
            _id = request.form['id']
            cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        try:
                cursor.execute("insert into
  job_apply(email,skill_score,edu_score,exp_score,job_id) "
                               "values(%s,%s,%s,%s,%s)", (email, skill, edu, exp, _id))
                mysql.connection.commit()
                cursor.execute("update job_post set application_count=application_count+1
  where job_id={}".format(_id))
                mysql.connection.commit()
                return jsonify(status="Successfully Applied..")
            except MySQLdb.IntegrityError:
                return jsonify(status="Already Applied")


  @app.route('/api/getJob/', methods=['POST'])
  def fetch_job():
        if request.method == "POST" and 'email' in request.form:
            email = request.form['email']
      else:
            return jsonify(status="error")
```

```
 @app.route('/api/userProfile/', methods=['POST'])
def get_profile():
    if request.method == "POST" and 'email' in request.form:
        email = request.form['email']


        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("SELECT * FROM candidate where email='{}'".format(email))
        user = cursor.fetchone()
        print(user)
        cursor.execute("SELECT * FROM candidate_edu where email='{}'".format(email))
        edu = cursor.fetchall()
        print(edu)
    cursor.execute("SELECT skill FROM candidate_skills where
email='{}'".format(email))
        skill = cursor.fetchall()
        print(skill)
        cursor.execute("SELECT * FROM candidate_exp where email='{}'".format(email))
        exp = cursor.fetchall()
        print(exp)
        return jsonify(user=user,skills=skill,exp=exp,edu=edu)


    else:
        return jsonify(status="error")


@app.route('/api/resume/', methods=['POST'])
def set_profile():


    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
  resume = request.files['file']
    email = request.form['email']
    fname = secure_filename(resume.filename)
    try:
        os.mkdir(os.path.join('./static/files/', email))
    except FileExistsError:
        print("Location Exists")
    resume.save(os.path.join('./static/files/' + email+"/", fname))
    reader = PdfReader(resume)
    print(len(reader.pages))
    page = reader.pages[0]
    text = page.extract_text()
    print(text)
    doc = nlp_ner(text)
    cursor.execute("delete from candidate_skills where email='"+email+"'")
    mysql.connection.commit()
    cursor.execute("delete from candidate_exp where email='"+email+"'")
  mysql.connection.commit()
    cursor.execute("delete from candidate_edu where email='"+email+"'")
    mysql.connection.commit()
    for ent in doc.ents:
        if ent.label_ == "SKILLS":
            try:
                cursor.execute("insert into candidate_skills(email,skill) values(%s,%s)", (email,
ent.text.lower()))
                mysql.connection.commit()
            except MySQLdb.IntegrityError:
                print("Exist Already")
        elif ent.label_ == "EDUCATION":
            try:
                cursor.execute("insert into candidate_edu(email,edu) values(%s,%s)",
(email,ent.text.lower()))
                mysql.connection.commit()
            except MySQLdb.IntegrityError:
```

```
print("Exist Already")
        elif ent.label_ == "EXPERIENCE":
            try:
                cursor.execute("insert into candidate_exp(email,exp) values(%s,%s)",
(email,ent.text.lower()))
                mysql.connection.commit()
            except MySQLdb.IntegrityError:
                print("Exist Already")
    cursor.execute("update candidate set resume =%s where email=%s",(fname,email))
    mysql.connection.commit()


    # sql = "INSERT INTO job_post(title, description, loc, sal,email) VALUES (%s, %s,
%s, %s,%s)"
    # values = (job_title, job_description, location, salary,email)
    # cursor.execute(sql, values)
    # mysql.connection.commit()
    # Return a JSON response with a success message
    response = {'status': 'success'}
    return jsonify(response)



@app.route('/api/c/reg/', methods=['POST'])
def c_signup():
    if request.method == 'POST' and 'email' in request.form:
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        email = request.form['email']
        password = request.form['pass']
        name = request.form['name']
        table = request.form['type']
        phone = request.form['phone']
        try:
            cursor.execute("insert into {} (email,password,name,phone)
values('{}','{}','{}','{}')".format(table,email,
             mysql.connection.commit()
            return jsonify(status="success")
        except MySQLdb.IntegrityError:
            return jsonify(status="error")
    else:
        return jsonify(status="error")



@app.route('/api/login/', methods=['POST'])
def login():
    if request.method == 'POST' and 'type' in request.form:
        email = request.form['email']
        password = request.form['pass']
  table = request.form['type']
        print(table)
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute("select * from " + table + " where email=%s and password=%s",
(email, password))
        data = cursor.fetchone()
        if data:
            return jsonify(status='success')
        else:
            return jsonify(status='error')
    else:
        return jsonify(status='error')



@app.route('/api/post_job', methods=['POST'])
def post_job():
    job_title = request.form['job_title']

    job_description = request.form['job_description']

location = request.form['location']

    salary = request.form['salary']

    email = request.form['email']

    cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)

    sql = "INSERT INTO job_post(title, description, loc, sal,email) VALUES (%s, %s, %s,
%s,%s)"

    values = (job_title, job_description, location, salary,email)

    cursor.execute(sql, values)

    mysql.connection.commit()

    cursor.execute("select job_id from job_post where email='{}' order by
post_time".format(email))

    job_id = cursor.fetchone()['job_id']

    job_description = job_description.replace("\n"," ")

    doc = nlp_ner(job_description)


    for ent in doc.ents:

        if ent.label_ == "SKILLS":

   try:

                cursor.execute("insert into job_skills(job_id,skill) values(%s,%s)", (job_id,
ent.text.lower()))

                mysql.connection.commit()

            except MySQLdb.IntegrityError:

                print("Exist Already")

        elif ent.label_ == "EDUCATION":

            try:

                cursor.execute("insert into job_edu(job_id,edu) values(%s,%s)", (job_id,
ent.text.lower()))

                mysql.connection.commit()

            except MySQLdb.IntegrityError:

                print("Exist Already")
        elif ent.label_ == "EXPERIENCE":
            try:
                cursor.execute("insert into job_exp(job_id,exp) values(%s,%s)", (job_id,
ent.text.lower()))
  mysql.connection.commit()
            except MySQLdb.IntegrityError:
                print("Exist Already")
    # Return a JSON response with a success message
    response = {'status': 'Job posted successfully.'}
    return jsonify(response)


@app.route('/api/viewPost/', methods=["POST"])
def get_req():
    if request.method == "POST" and 'email' in request.form:
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        email = request.form['email']
        cursor.execute("select * from job_post where email = '" + email + "' order by
post_time desc")
        data = cursor.fetchall()
        return jsonify(data)
    else:
        return jsonify(status="Request error")


if _name_ == '_main_':
    app.run(host='0.0.0.0', port=5000)
```

NER Model Training

```
import spacy
from spacy.tokens import DocBin
from tqdm import tqdm



import json
def read_jsonl(filename):
  DATA = []
  labeled_data = []
  with open(filename, "r") as read_file:
      for line in read_file:
          data = json.loads(line)
          labeled_data.append(data)
  for entry in labeled_data:
      entities = []
      for e in entry['label']:
          entities.append((e[0], e[1],e[2]))
      spacy_entry = (entry['text'], {"entities": entities})
DATA.append(spacy_entry)
  return DATA

TRAINING_DATA = read_jsonl("/content/drive/MyDrive/dataset/Job Portal/train.jsonl")
TEST_DATA = read_jsonl("/content/drive/MyDrive/dataset/Job Portal/test.jsonl")

print(TRAINING_DATA[0])
```

```
('IDR is an employee-owned staffing firm that puts people first. We go above and beyond
by empowering talent and developing leaders to deliver best-in-class solutions. Founded
```

and headquartered in Atlanta, we’ve grown organically since 1998 to serve businesses and professionals throughout the Nashville, Dallas, Fort Worth, Denver, and Birmingham markets. We have a prestigious client looking for a Sr. Java Developer Java Developer Job Duties: Defines site objectives by analyzing user requirements; envisioning system features and functionality. Designs and develops user interfaces to internet/intranet applications by setting expectations and features priorities throughout development life cycle; determining design methodologies and tool sets; completing programming using languages and software products; designing and conducting tests. Recommends system solutions by comparing advantages and disadvantages of custom development and purchase alternatives. Integrates applications by designing database architecture and server scripting; studying and establishing connectivity with network systems, search engines, and information servers. Creates multimedia applications by using authoring tools. Completes applications development by coordinating requirements, schedules, and activities; contributing to team meetings; troubleshooting development and production problems across multiple environments and operating platforms. Supports users by developing documentation and assistance tools. Updates job knowledge by researching new internet/intranet technologies and software products; participating in educational opportunities; reading professional publications; maintaining personal networks; participating in professional organizations. Enhances organization reputation by accepting ownership for accomplishing new and different requests; exploring opportunities to add value to job accomplishments. REQUIRED 5+ years of industry experience focusing on Backend development 3+ years of experience working with Java/Spring Boot framework 3+ years of experience working with Go, NodeJS , and/or Python 3+ year of experience in an AWS environment Hands-on experience developing server-side applications and API’s Experience designing and implementing architectural patterns, specifically microservices Unit testing and TDD best practices with tools like Junit, Mockito, Spock Experience with CI/CD , using SCM tools such as Git as well as an understanding of Gitflow Hands on experience with SQL and NoSQL (e.g. Dynamo, Mongo, etc.) database technologies', {'entities': [(402, 407, 'SKILLS'), (417, 422, 'SKILLS'), (1919, 1950, 'EXPERIENCE'), (1983, 2005, 'EXPERIENCE'), (2019, 2023, 'SKILLS'), (2024, 2035, 'SKILLS'), (2046, 2068,

```
'EXPERIENCE'), (2082, 2084, 'SKILLS'), (2086, 2093, 'SKILLS'), (2102, 2109,
'SKILLS'), (2109, 2130, 'EXPERIENCE'), (2294, 2308, 'SKILLS'), (2308, 2320,
'SKILLS'), (2360, 2365, 'SKILLS'), (2367, 2374, 'SKILLS'), (2430, 2434, 'SKILLS'),
(2498, 2502, 'SKILLS'), (2506, 2512, 'SKILLS'), (2518, 2524, 'SKILLS'), (2526, 2531,
'SKILLS')]})
```

```
nlp = spacy.blank("en") # load a new spacy model
db = DocBin() # create a DocBin object

def db_to_disk(DATA,filename):
for text, annot in tqdm(DATA):
    doc = nlp.make_doc(text)
    ents = []
    for start, end, label in annot["entities"]:
        span = doc.char_span(start, end, label=label, alignment_mode="contract")
        if span is None:
            print("Skipping entity")
        else:
            ents.append(span)
    doc.ents = ents
    db.add(doc)
  db.to_disk(filename) # save the docbin object

db_to_disk(TRAINING_DATA,"./train_data.spacy")
db_to_disk(TEST_DATA,"./test_data.spacy")
 32%|███       | 32/99 [00:00<00:00, 108.28it/s]Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
 68%|██████▊   | 67/99 [00:00<00:00, 143.53it/s]Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
Skipping entity
100%|██████████| 99/99 [00:00<00:00, 143.18it/s]
Skipping entity
Skipping entity
Skipping entity
100%|██████████| 10/10 [00:00<00:00, 248.14it/s]
Skipping entity


! python -m spacy init config config.cfg --lang en --pipeline ner --optimize efficiency
2023-04-19 04:28:44.148473: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
⚠ To generate a more effective transformer-based config (GPU-only),
install the spacy-transformers package and re-run this command. The config

command. The config

generated now does not use transformers.

ℹ Generated config template specific for your use case

- Language: en

- Pipeline: ner

- Optimize for: efficiency

- Hardware: CPU

- Transformer: None

✔ Auto-filled config with all values

✔ Saved config

config.cfg

You can now add your data and train your pipeline:

python -m spacy train config.cfg --paths.train ./train.spacy --paths.dev ./dev.spacy


! python -m spacy train config.cfg --output ./ --paths.train ./train_data.spacy --paths.dev
./test_data.spacy

2023-04-19 04:29:59.780737: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-
TRT Warning: Could not find TensorRT

ℹ Saving to output directory: .

ℹ Using CPU


=========================== Initializing pipeline
===========================

[2023-04-19 04:30:02,724] [INFO] Set up nlp object from config

[2023-04-19 04:30:02,738] [INFO] Pipeline: ['tok2vec', 'ner']

[2023-04-19 04:30:02,744] [INFO] Created vocabulary

[2023-04-19 04:30:02,745] [INFO] Finished initializing nlp object

[2023-04-19 04:30:06,084] [INFO] Initialized pipeline components: ['tok2vec', 'ner']

✔ Initialized pipeline
```

```
============================= Training pipeline
=============================

ℹ Pipeline: ['tok2vec', 'ner']

ℹ Initial learn rate: 0.001

E    #       LOSS TOK2VEC  LOSS NER  ENTS_F  ENTS_P  ENTS_R  SCORE

-- ------- -------- ------. ------- ---- ---- ----

 0     0         0.00    475.79    0.00    0.00    0.00    0.00

 2    200      1527.06   9658.88   78.93   79.65   78.23    0.79

 4    400       281.15   1607.38   84.93   83.05   86.89    0.85

 6    600       274.52   1041.60   90.87   92.99   88.85    0.91

 8    800       195.53    760.46   91.31   95.41   87.54    0.91

 10   1000      1588.72    683.74  91.00   95.03   87.31

0.91

 12   1200       648.57    566.51   92.95   94.43   91.52    0.93

 14   1400       264.93    457.31   94.14   96.45   91.93    0.94

 16   1600       283.53    392.83   94.07   95.93   92.29    0.94

 18   1800       293.68    382.67   94.98   96.34   93.65    0.95

 20   2000       299.46    372.23   93.78   95.06   92.53    0.94

 22   2200       281.55    332.19   94.02   95.36   92.70    0.94

 24   2400       342.34    347.89   94.35   95.62   93.12    0.94

 26   2600       325.87    290.18   94.06   97.22   91.10    0.94

 28   2800       473.78    301.34   94.77   94.99   94.54    0.95

30   3000       400.48    282.97   94.95   95.72   94.19    0.95

 32   3200      1043.13    275.82   94.77   96.10   93.48    0.95

 34   3400       501.96    294.84   94.62   94.82   94.42    0.95

✔ Saved pipeline to output directory

model-last


!zip -r model.zip model-best/


zip error: Nothing to do! (try: zip -r model.zip . -i model-best/)
```

Load the trained model

```
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive



import spacy
from spacy.tokens import DocBin
from tqdm import tqdm



!unzip "/content/drive/MyDrive/dataset/Job Portal/model.zip"
Archive: /content/drive/MyDrive/dataset/Job Portal/model.zip
   creating: model-best/
  inflating: model-best/config.cfg
 creating: model-best/tok2vec/
 extracting: model-best/tok2vec/cfg
 inflating: model-best/tok2vec/model
 creating: model-best/ner/
  inflating: model-best/ner/cfg
  inflating: model-best/ner/model
  inflating: model-best/ner/moves
  creating: model-best/vocab/
extracting: model-best/vocab/vectors.cfg
 extracting: model-best/vocab/key2row
 inflating: model-best/vocab/strings.json
 inflating: model-best/vocab/vectors
 extracting: model-best/vocab/lookups.bin
```

```
    inflating: model-best/tokenizer

    inflating: model-best/meta.json


  nlp_ner = spacy.load("/content/model-best")


  txt = '''


As a Software Engineer, you need to contribute to software product developed in global team. Being a part of the development team, you will contribute to end to end software development. This will include working on software deliverable for the release, meeting required design objectives to meet customer & business needs.GE Healthcare is a leading global medical technology and digital solutions innovator.

Our mission is to improve lives in the moments that matter. Unlock your ambition, turn ideas into world-changing realities, and join an organization where every voice makes a difference, and every difference builds a healthier world.


Job Description


Roles and Responsibilities


In this role, you will:


Be responsible for programming a feature and or a small feature set

Translate requirements / vision into prioritized list of user stories and deliver to required timelines and quality standardsContribute to the team and works with direction from senior team members or management, beginning to work independently.

Analyse, design and Implement the Business

Requirements.
```

Apply principles of Software Development Life Cycle and methodologies like Lean/Agile/XP, CI, Software and Product Security, Scalability, Documentation Practices, refactoring and Testing Techniques.

Analyse, prioritize & resolve the production issues.

Be responsible for troubleshooting customer issues and providing necessary resolution by understanding and applying GE processes.

Document unit test plans, executing unit testing successfully.

Deliver complex software components on schedule with quality.

Perform technology/tools evaluation and software technical proof of concepts.

Design and develop optimized and reliable code as per the specifications.

Write code that meets standards and delivers desired functionality using the technology selected for the project

Work with Product owner, Architect, Technical lead, Scrum master to understand product requirements & vision

Apply principles of SDLC and methodologies like Lean/Agile/XP, CI, Software and Product Security, Scalability, Documentation Practices, refactoring and Testing Techniques

Proactively share information across the team, to the right audience with the appropriate level of detail and timeliness

Education Qualification

Bachelor's Degree in Computer Science or “STEM” Majors (Science, Technology, Engineering and Math)

Technical Expertise:

```
4+ years of experience on designing and developing applications using C/C++ / Core JAVA technologies,

has participated in all aspects of the Software Development Life Cycle (SDLC).

Knowledge of data structures, multithreading, concurrency, collections.

Skilled in breaking down problems, documenting problem statements and estimating efforts

Problem solving skills to assess design alternatives and perform trade-offs to determine designs or functionality that best meet needs.

Experience in SDLC documentation to comply with ISO and European standards.

Knowledge of Object-Oriented Analysis and Design, Software Design Patterns.

Demonstrates the initiative to explore alternate technology and approaches to solving problems

Demonstrates awareness about competitors and industry trends

Has the ability to analyze impact of technology choices

Desired Skills:



Experience with Windows and Unix / Linux based programming and Scripting skills (Shell, Python, Bash, Perl)

Understanding of Serial protocols, Embedded development

Knowledge of unit testing using tools such as Junit, CPPUnit

Experience working within Agile/Scrum software development framework

Debugging and Troubleshooting Expertise

Experience working on software projects in the Healthcare domain

'''



    doc = nlp_ner(txt)
```

```
spacy.displacy.render(doc, style="ent", jupyter=True) # display in Jupyter


!pip install PyPDF2

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/

Collecting PyPDF2

  Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)

     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━

━━━━━━━ 232.6/232.6 kB 10.1 MB/s eta 0:00:00

Requirement already satisfied: typing_extensions>=3.10.0.0 in /usr/local/lib/python3.9/dist-packages (from PyPDF2) (4.5.0)

Installing collected packages: PyPDF2

Successfully installed PyPDF2-3.0.1


from PyPDF2 import PdfReader



reader = PdfReader("/content/resume.pdf")

print(len(reader.pages))

page = reader.pages[0]

text = page.extract_text()

print(text)

2

Gokul G Menon

/githubGokul-GMenon |/linkedingokul-g-menon |/envel⌢pegokulgovindmenon@gmail.com

Education

2020 - present BTech in Computer Science and Engineering at College Of Engineering Trivandrum

(CGPA: 9.30/10.0)

2019 Class 12th CBSE (95% PCM)

Work Experience
```

AI intern at Navana.ai February 2023 - April 2023

Worked on projects using OpenAI APIs, optimization using parallel processing, python

script generation, preparing and running model evaluations and inference scripts etc.

(Python, PyTorch, Pandas, HuggingFace)

Computer Vision Intern at Quetzal Robotics July 2022 - Sept 2022

Trained object detection models on YOLOv5, YOLOv3 and worked on implementing such models into a collective package that can track objects.

(Deep Learning, TensorFlow, YOLO, OpenCV)

Intern at Wizard flair PVT LTD, Kochi, Kerala July 2021 - Oct-2021

Worked on generating posters on demand for businesses using minimal input data, internal website for

sorting internal tools in one place using Django, caption generation for the posters generated using user

info (Python, Django, MySQL, HTML, CSS, BootStrap)

Standout Projects

Publication on Automatic Recognition of Continuous Malayalam Speech using Pretrained

Multilingual Transformers accepted for ICISCoIS 2023

conference Click here for the paper

Click here for viewing XLS-R work and here for the Ken-LM Language Model work

Paper on Malayalam ASR accepted by ICISCoIS 2023 conference (Awaiting publication on IEEE Explore).

Fine tuned Whisper Language Model for the Malayalam language Click here for the project

Fine tuned the Facebook Whisper speech recognition model on the Malayalam language and published

on hugging face

XLS-R model for Malayalam language used to build translator API

Click here for the project

The XLS-R model fine tuned on the Ken-LM language model was used to build a Malayalam to English

language audio translator as a Django API

Build an open source Resume Analyzer as part of FOSS Hack using OpenAI API Click here for the project

Built a Resume Analyzer for recruiters to group

resumes as per required skill sets. In the team, I

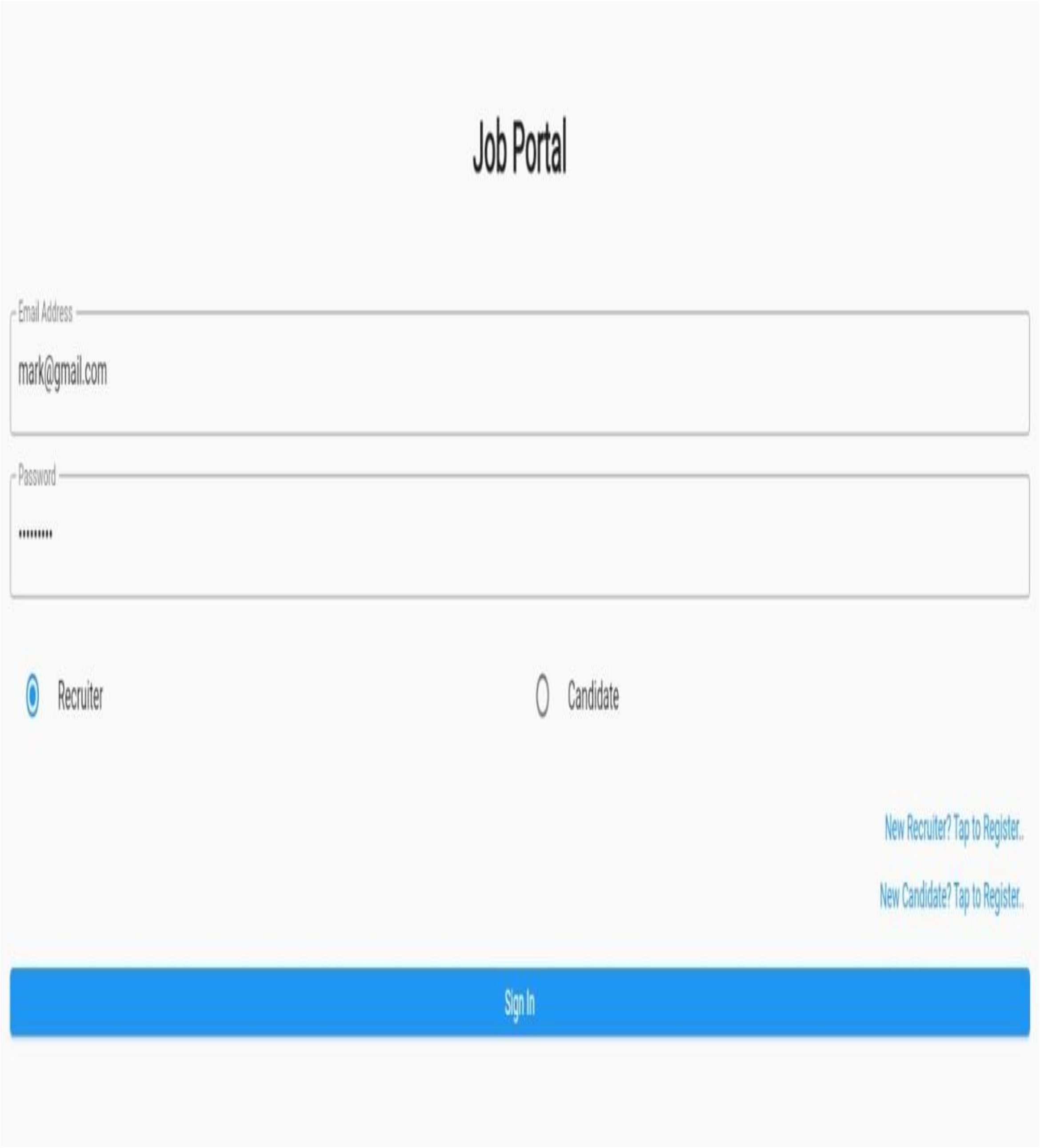
contributed towards converting PDF of resumes to text, extracting information regarding presence of

required skills using OpenAI API, hosting the model as a Django API etc. Planning to expand by

training a model to detect the skill set information from resumes

```
doc = nlp_ner(text)
spacy.displacy.render(doc, style="ent", jupyter=True) # display in Jupyter
```

SCREENSHOTS

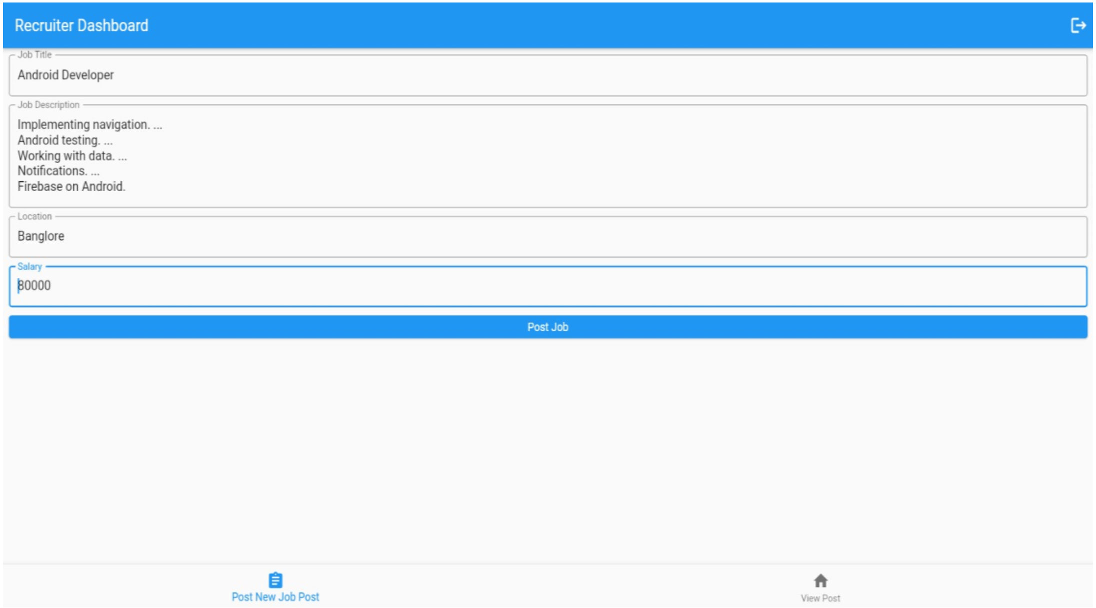


Job Portal
Sign In
New Recruiter? Tap to Register..
New Candidate? Tap to Register..

Email Address: mark@gmail.com

Password: ••••••••

● Recruiter ○ Candidate



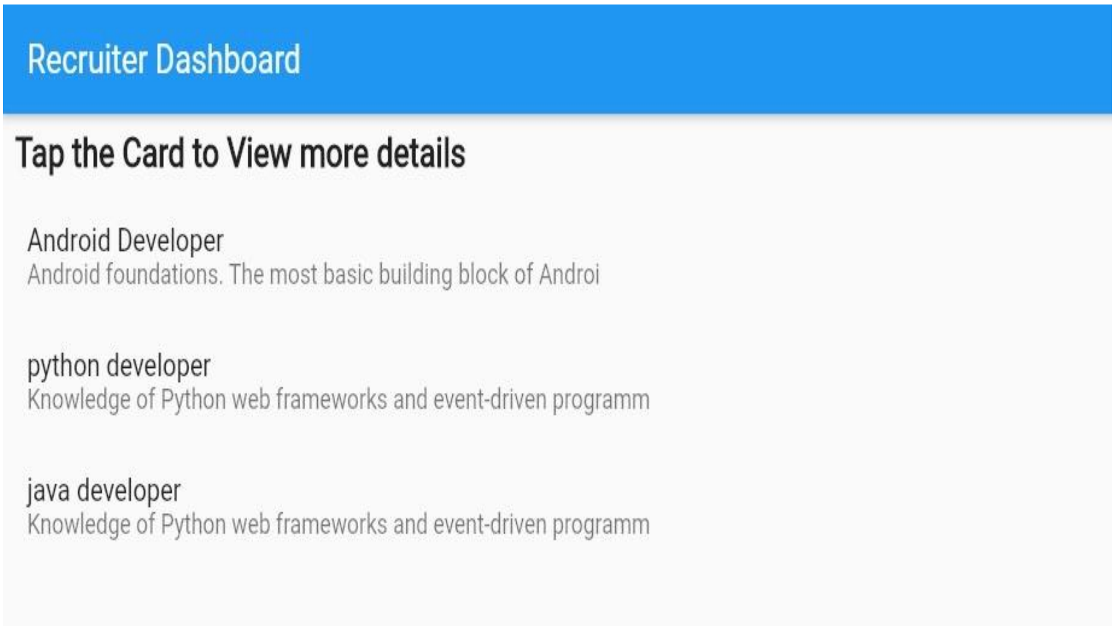
Recruiter Dashboard

Job Title
Android Developer

Job Description
Implementing navigation. ...
Android testing. ...
Working with data. ...
Notifications. ...
Firebase on Android.

Location
Banglore

Salary
80000



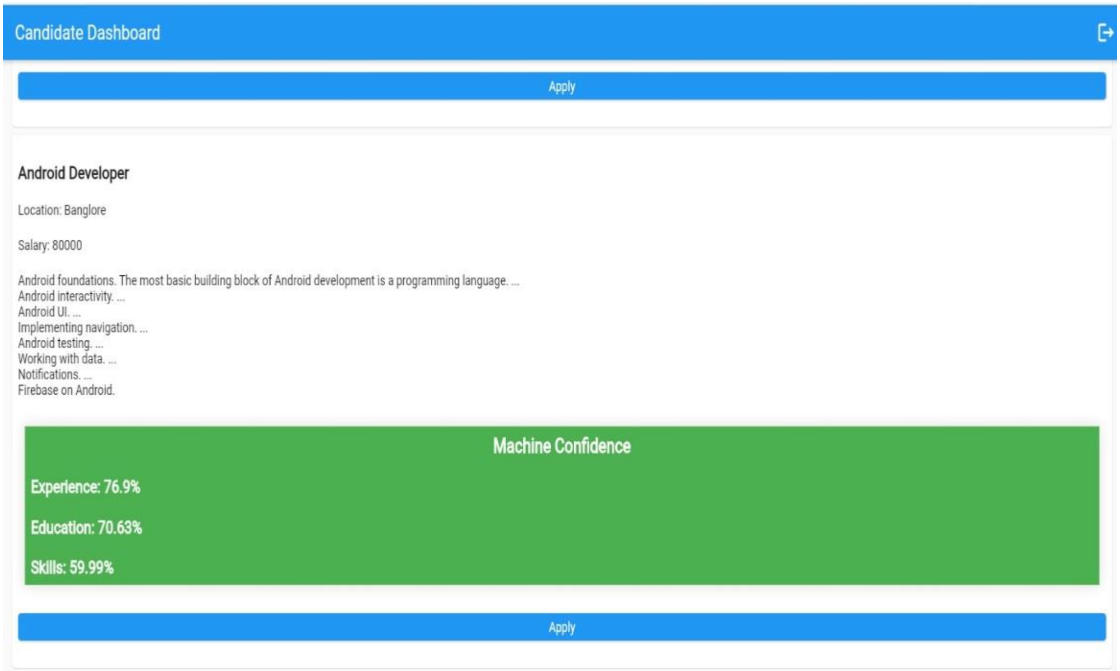
Recruiter Dashboard

Tap the Card to View more details

Android Developer
Android foundations. The most basic building block of Androi

python developer
Knowledge of Python web frameworks and event-driven programm

java developer
Knowledge of Python web frameworks and event-driven programm



Android Developer

Location: Banglore

Salary: 80000

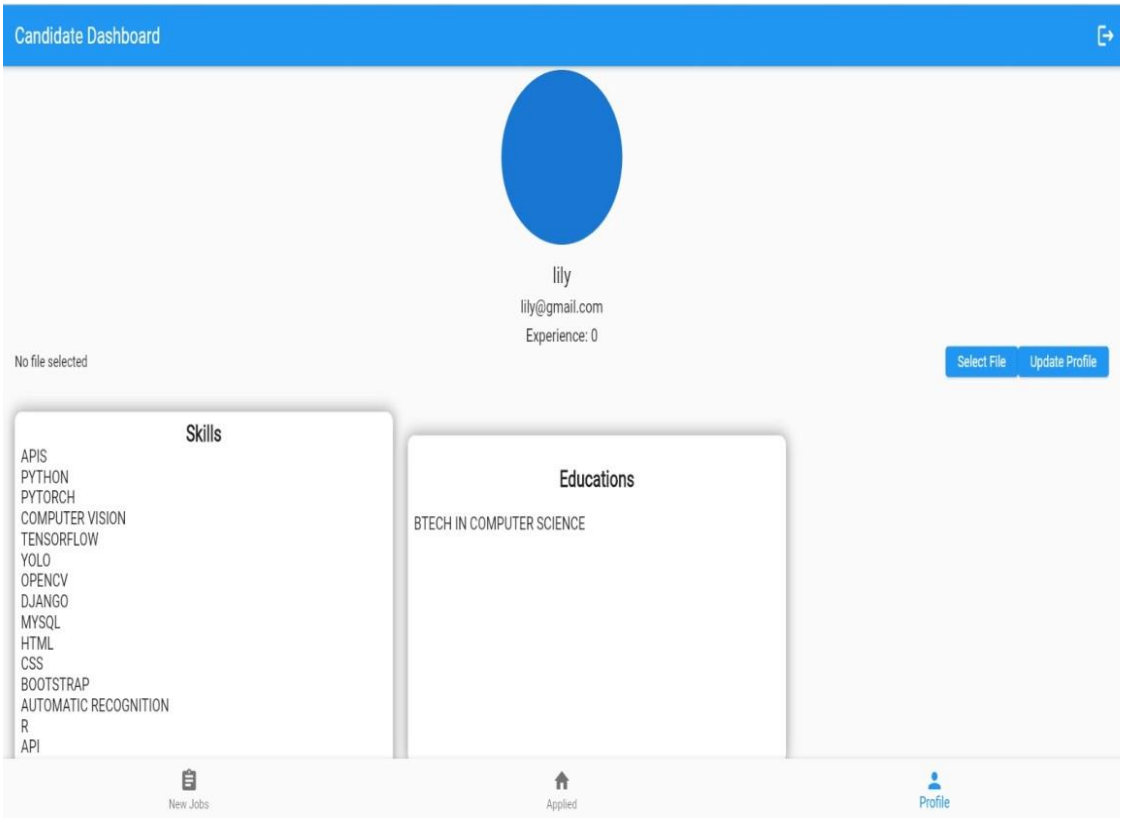
Android foundations. The most basic building block of Android development is a programming language. ...
Android interactivity. ...
Android UI. ...
Implementing navigation. ...
Android testing. ...
Working with data. ...
Notifications. ...
Firebase on Android.

Machine Confidence

Experience: 76.9%

Education: 70.63%

Skills: 59.99%

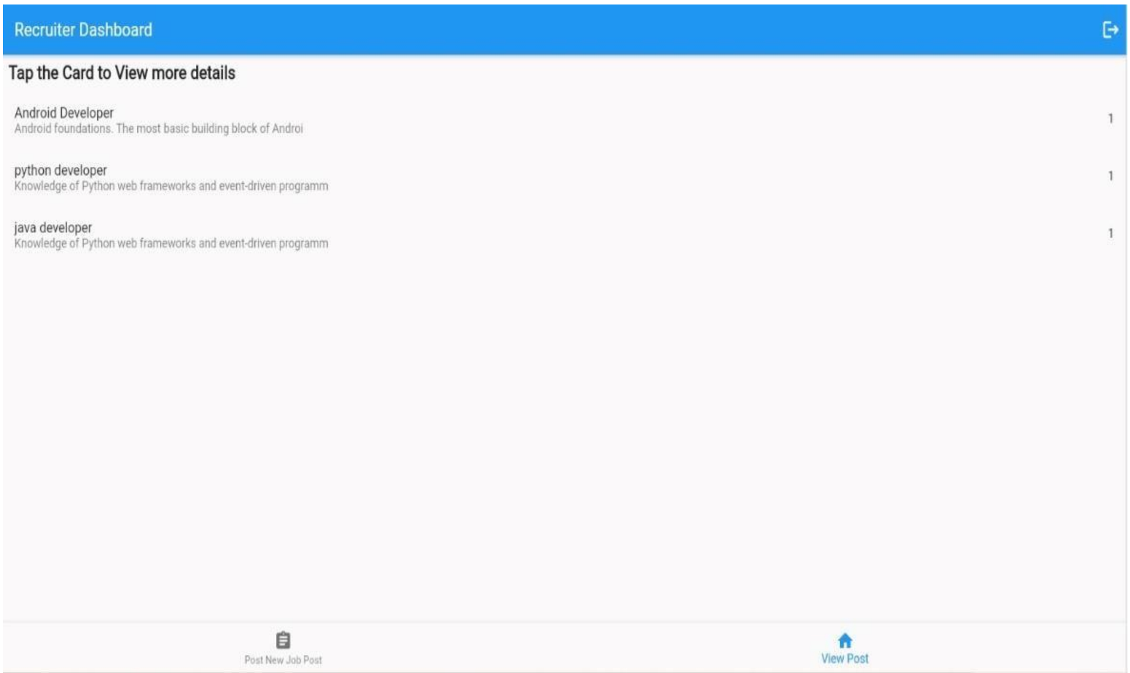


lily
lily@gmail.com
Experience: 0

No file selected

Skills
APIS
PYTHON
PYTORCH
COMPUTER VISION
TENSORFLOW
YOLO
OPENCV
DJANGO
MYSQL
HTML
CSS
BOOTSTRAP
AUTOMATIC RECOGNITION
R
API

Educations
BTECH IN COMPUTER SCIENCE



Recruiter Dashboard

Tap the Card to View more details

Android Developer 1
Android foundations. The most basic building block of Androi

python developer 1
Knowledge of Python web frameworks and event-driven programm

java developer 1
Knowledge of Python web frameworks and event-driven programm

REFERENCES

[1] Suhas H E,Manjunath A, "Differential Hiring using a Combination of NER and Word Embedding ", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878 (Online), Volume-9 Issue-1, May 2020

[2] Jaichandran R , Leelavathy S , Kanaga Suba Raja , Pranav Kumar , Sailendra Kumar Majhi1 and Vikas Singh Thakur, " JOB PORTAL RESUME EVALUATON SYSTEM USING TEXT MINING AND NATURAL LANGUAGE PROCESSING " European Journal of Molecular & Clinical Medicine ISSN 2515-8260 Volume 7, Issue 4, 2020

[3] Suleiman Ali Alsaif, “NLP-Based Bi-Directional Recommendation System: Towards Recommending Jobs to Job Seekers and Resumes to Recruiters", International Journal of Computer Applications, Vol. 178, No. 37, December 2018.

[4] Hang LI, Special Section on Information-Based Induction Sciences and Machine Learning A Short Introduction to Learning to Rank, IEICE TRANS. INF. & SYST., VOL.E94–D, NO.10 OCTOBER 2011

[5] Article by Ideal: Effective recruiting using AI, Article link.

[6]. Ayishathahira C H, Sreejith C, Raseek C. Combination of Neural Networks and Conditional Random Fields for Efficient Resume Parsing. 18233630