RESUME RANKING SYSTEM USING MACHINE LEARNING &NLP

Prof. **Prof. Dharamveer Singh Sisodiya[1]**

Harmanpreet Singh[2], Ketan Mishra[3], Parameswar Das[4], Anupam Sagar[5]
Computer Engineering Department, ISBM College of Engineering Nande, Pune-412115,
India Savitribai Phule Pune University

ABSTRACT:

Selecting desirable applicants for an organization is one of the major challenges in human resource management. Through e-recruitment, candidates can now be represented on the web under a wider range of possibilities. This manual screening process may hinder the team's efforts to find the right candidate at the right time. The difficult screening process may be significantly streamlined by using an automated system for screening and ranking applicants. To accommodate the identified intelligence in hiring patterns, machine learning would be applied to understand the intelligence behind them. Using ML (Machine Learning) to rank the resumes with the given job requirements to match the best comparable candidates which is the most important and crucial task for any company to hire an ideal candidate for their job role. To classify resumes based on their respective categories, NLP (Natural Language Processing) and Machine Learning (ML) are used, while MaLSTM model(Siamese networks + LSTM with Manhattan distance) is used to rank candidates based on their resume's similarity to the job description.

I. INTRODUCTION

In the IT business, evaluating resumes is an important step for finding new candidates with the required skills. Thousands of resumes are review by HR team. People come from different fields of profession and have different backgrounds. Each one of them has had different types of education, has worked on different projects and thus has a unique style of presenting his/her credentials in the resume. Resumes are unstructured documents that come in various file formats (.pdf, .doc, .docx, .jpg, .txt etc.) and their content is not written according to standard formats or templates. This means reading resumes is not simple and thus recruiters spend a large amount of time going through the resumes for selecting the right candidates. These resumes will be automatically processed by the information extraction system. Extracted information such as name, phone/contact details, emails id's, qualification, experience, skill-sets etc. can be stored as a structured data in a DB and then can be used in various different areas/fields.

This project enables recruiters to sort through a sea of resumes to locate the best applicants who fulfil the job requirements (particularly during high-volume recruiting). Applications are filtered based on qualifications for open positions, including experience, education, and skills. the end result of scoring each CV based on an intelligent comparison to the relevant Job Description. The project automates the process using machine learning and natural language processing technique.

II. PROBLEM DEFINITION

Presently, the company's recruitment department has to review each and every resume for a particular employment post, which is typically time-consuming. Sometimes a deserving candidate loses their opportunity because the recruitment receives a large volume of resumes. Therefore, this research paper develop a system that will aid the recruitment team in sorting relevant applications for certain job roles using Natural Language Processing (NLP) and other machine learning concepts, and then score the resumes based on experience and other factors.

III. LITERATURE SURVEY

Professor Dr.K.Satheesh and , A.Jahnavi proposed a system using advanced Natural Language Processing which is a field in Machine Learning.Model helps the recruiters in screening the resumes based on job description with in no time. It makes the hiring process easy and efficient by extracting the required entities automatically by using Spacy NER model

Professor Ashif Mohamed proposed a system using Ontology where we can compare the resume models with the given job requirements to match the best comparable candidates. Two ranking algorithms are underlined in this system which will be invoked to assign a ranking point to the recommended candidates against the other candidates on the recommendation pool.

Professor Sayed Zainul Abideen Mohd Sadiq and Juneja Afzal Ayub Designed an automated system to extract information from unstructured resumes and transform that information to structured format. And ranking those resumes based on the information extracted, according to the skill sets of the candidate and based on the job description of the company.

IV. ASSUMPTIONS AND DEPENDENCIES

Assumptions:

The following Assumption was taken into consideration:

• By parsing resumes, the parser converts them into text files, which are then read to retrieve the attributes of the candidates and store them in structured form in the JSON files.

• This json file contains the attributes of the students in ranked format and then it is shown as the output to the student/employer.

Dependencies:

The dependencies are as follows:

• For User interface node.js and react.js web framework is used.

• Python programming language, NLTK, NumPy, Matplotlib, and TensorFlow are used to parse the document and store the information in structured form.

• To get user information form LinkedIn, GitHub there api is used which provides data in structured format.

V. SYSTEM ARCHITECTURE

This approach is utilized with a variety of technologies in mind, with the user always coming first. Our main goal is to create a system that is durable and survives the test of time, which also aids in the product's scalability and the user's benefit.

For the application's UI, we have used node.js and react. Any inexperienced user can make efficient use of the User Interface because it is reliable and actually fairly easy.

Our system follows the three tier architecture . First tier consist of GUI, Processing block and the Database.

GUI:

In our project, the GUI (Graphical User Interface) is responsible for the user interface where users upload their resumes in any format (PDF, DOC, DOCX, etc.) and links to their social media accounts. The GUI provides a platform for the user to communicate with the database. It serves as a communicator and connector that links the database and facilitates data flow between the GUI and the database.
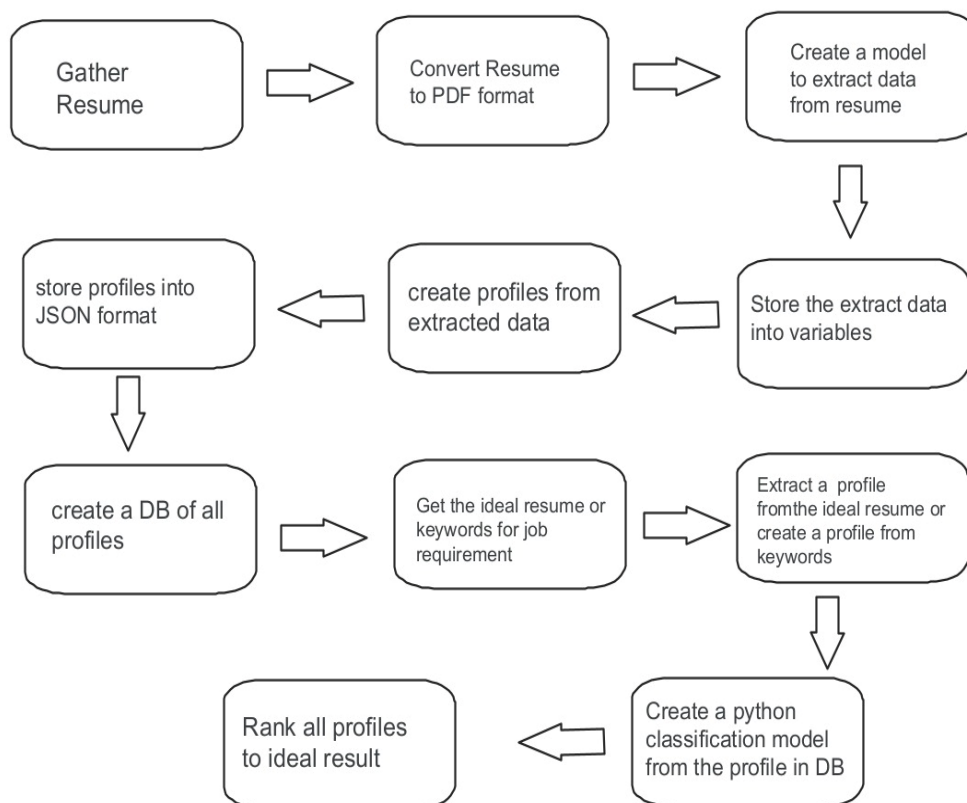
Processing block:

The processing block is the one where our project is actually processed. This block links the GUI to the database, serving as both a connector and a communicator to link the two together and facilitate data flow between the GUI and the database. Its major purpose is to parse information from resumes and social media profiles of candidates and store it in a database in an organized format (json). After storing this information this system will give output using web application.

Database:

Database tier is the tier used for the storage of data. All the information required for processing the entire project is in this tier. The information in this tier relates to the student data that was gleaned from their resumes and social media sites.

VI. SYSTEM DESIGN

- The initial step is to gather resumes from global websites like Kaggle, github, and linkedin.
- There are various different file formats for resumes such as pdf, .doc, .docx, .jpg, .txt etc. All resumes will be converted to PDF format.
- Build a model to extract information from resumes like skills, phone number, email, and etc.
- Store the extracted data in variables, then use the extracted data to build a profile. Create a database of all the profiles after storing them in JSON format. From there, you can get the best resume or keywords for the position.
- Build a Python model from the database profile using the ideal resume as a template. compares all profiles to the ideal profile and ranks them.

```
┌──────────┐      ┌──────────────┐      ┌──────────────┐
│ Gather   │  ⇒   │ Convert Resume│  ⇒   │ Create a model│
│ Resume   │      │ to PDF format │      │ to extract data│
│          │      │              │      │ from resume  │
└──────────┘      └──────────────┘      └──────────────┘
                                                 ⇓
┌──────────────┐   ┌──────────────┐     ┌──────────────┐
│ store profiles│ ⇐ │ create profiles│ ⇐ │ Store the extract│
│ into JSON     │   │ from extracted │   │ data into    │
│ format        │   │ data          │   │ variables    │
└──────────────┘   └──────────────┘     └──────────────┘
      ⇓
┌──────────────┐   ┌──────────────┐     ┌──────────────┐
│ create a DB  │ ⇒ │ Get the ideal │ ⇒  │ Extract a profile│
│ of all       │   │ resume or     │    │ from the ideal  │
│ profiles     │   │ keywords for  │    │ resume or create│
│              │   │ job requirement│   │ a profile from │
│              │   │               │    │ keywords        │
└──────────────┘   └──────────────┘     └──────────────┘
                                                 ⇓
         ┌──────────────┐        ┌──────────────┐
         │ Rank all     │   ⇐    │ Create a python│
         │ profiles to  │        │ classification │
         │ ideal result │        │ model from the │
         │              │        │ profile in DB  │
         └──────────────┘        └──────────────┘
```

VII. ANALYSIS MODEL: SDLC MODEL TO BE APPLIED

PHASE 1 – PLANNING AND ANALYSIS

According to the problem statement given we will plan the requirements and process. And even analyse the planned structure for any future error.

1.1 DATA CLEANING

- Correcting mistakes such as empty fields
- Identifying duplicate data points

1.2 **DATA GATHERING**

The data gathered has been taken from Kaggle which we can fetch using an api or we can use pandas lib to import data from the Kaggle database to pre process it in next step.

PHASE 2 – DESIGNING THE PRODUCT

In this phase we will design the product according to the plan made earlier.

PHASE 3 – DEVELOPING AND CODING

Here we will write the code as per the requirements and define methods.

- Parse resume in varying formats (.docx, .doc, .pdf, .txt, etc) to convert and store in a database.
- Build a Model such that it should be able to extract ROLES, SKILLS, PERIOD, EDUCATION, COMPANIES, besides the usual attributes of NAME, EMAIL, PHONE number and store it in database
- The Model should also be able to build/ synthesize/ extract a synopsis of the Candidate's profile.

PHASE 4 – TESTING

In testing we will test the various test cases like for example- we will import resumes in various other formats and check that the system is providing the appropriate job as per one profile.

PHASE 5 – MAINTENANCE

Here we will keep updating the system with various features to sort and rank resumes efficiently.

VIII. SOFTWARE AND HARDWARE REQUIREMENTS

- **Language used** : Python (version 3.9 and above)

- **Software Requirements** *:* Jupyter Notebook

- **Hardware Requirements** :

  - Processor-Intel(R) Core(TM) i5-6440HQ CPU @ 2.60GHz   2.59 GHz
  - Installed RAM-8.00 GB (7.88 GB usable)
  - System type- 64-bit operating system, x64-based processor

IX. OTHER SPECIFICATION

ADVANTAGES

- The deserving candidates would get an opportunity and their skills would be appreciated.

- The process reduces the time consumption of the recruitment team for hiring process.

- We have build such a model which will convert all the document formats into single TEXT format.

LIMITATIONS

- The Above systems has models that don't have any way to improve themselves over the time, the models will be trained only once.

- The above models used Machine learning algorithms which have a tend to plateau in performance when runned over a large dataset.

X. CONCLUSION

A benchmarking is done to observe the results obtained by the manual process and the resumes recommended by the proposed method through Machine Learning. The results obtained by the proposed method are mathematically and practically much better than traditional methods. Ranking and Re-ranking based on the Hiring Pattern are very useful for next generation Head Hunting solution.

XI. ACKNOWLEDGMENT

XII. REFERENCES

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

[2]   Identifying ''best'' applicants in recruiting using data envelopment analysis Sharon A. Johnson, JoeZhu. http://www.sciencedirect.com/science/article/pii/S0038012102000484

[3]   Referenced Links : Jessica Simko , "How Hiring Managers Make Decisions" http://www.careerealism.com/hiring-managers-decisions/ Vinayak Joglekar , "Ranking Resumes using MachineLearning" https://vinayakjoglekar.wordpress.com/2014/06/24/ranking-resumes-

using-machine Peter Gold , "Artificial Intelligence Recruiting" https://www.linkedin.com/pulse/artificial-intelligence-recruiting-peter-gold Turbo Ricruit
,"Automated Application Processing","Better candidate experience","Matching Job Descriptions to Resumes" http://www.turborecruit.com.au/benefits-ofartificial-inteligence-for-recruitment/

[4] Referenced :- Professor Dr.K.Satheesh and , A.Jahnavi proposed a system using advanced Natural Language Processing which is a field in Machine Learning

[5] Referenced :-   Professor Ashif Mohamed proposed a system using Ontology where we can compare the resume models with the given job requirements to match the best comparable candidates

[6] Referenced :- Professor Sayed Zainul Abideen Mohd Sadiq and Juneja Afzal Ayub Designed an automated system to extract information