

A Mini Project Report on

Resume Parser for ATS with Job Role Recommendation

Submitted in partial fulfilment for the degree of
degree of Bachelor of Technology in Data Science

Submitted by
Yashvi Ghumre (15)
Priyanka Lamkoti (27)

Under the guidance of
Dr. Santoshi Pote



Department of Data Science

USHA MITTAL INSTITUTE OF TECHNOLOGY
S.N.D.T WOMEN'S UNIVERSITY
MUMBAI – 400049

April, 2024.

CERTIFICATE

This is to certify that **Yashvi Ghumre (15), Priyanka Lamkoti (27)** have completed the Minor Project Phase-II work on the topic "**Resume Parser for ATS with Job Role Recommendation**" satisfactorily in the partial fulfillment for the Bachelor's Degree in **Data Science** during the year 2023-24 as prescribed by S.N.D.T. Women's University.

Guide

Dr. Santoshi Pote

H.O.D

Prof. Rajesh Kolte

Principal

DR. Yogesh Nerkar

Examiner 1

Examiner 2

ACKNOWLEDGEMENT

We would like to express our sincere appreciation to everyone who gave their invaluable assistance and support in this project. We are highly indebted to Dr. Santoshi Pote for her guidance and constant supervision as well as for providing necessary information regarding the project also for her support in this project. Without her help, this project would not have been possible.

We would also like to extend our gratitude to our classmates who provided valuable feedback and encouragement throughout the process. Finally, we would like to express our appreciation to our families for their unwavering support and understanding during the long hours and late nights spent working on this project.

Thank you all for your contributions, support, and encouragement.

0.1 Abstract

In recent times, due to the tremendous amounts of applications on the daily, job recruiters and hiring managers claim to spend only 6-7 seconds glossing over a resume before they decide if a candidate is fit for the position or not. The very initial resume screening, however, can be completely automated using Applicant Tracking Systems.

This project aims to build a resume parser that first uses Natural Language Processing to scan for information that are relevant to the recruiters. It will also involve creating a database of all the applicants giving easy access to the recruiters as and when necessary. The next part will then be to build a system to match the parsed information with several job descriptions to see where exactly the candidate fits best, if not the position they applied for.

The goal is to evaluate the candidates with their qualifications, experience, and skill profiles. And keeping a record of the applicants for future references and openings, while also prioritizing efficiency, accuracy and, fairness in an organization's hiring process.

Contents

0.1	Abstract	1
1	Introduction	4
1.1	Problem Statement	5
2	Block Diagram	6
3	Literature Survey	8
3.1	Review of Papers	8
4	Methodology	12
4.1	Research Design	12
4.2	Data Analysis	12
4.3	Ethical Considerations	13
5	Results and Discussions	14
6	Conclusion	16
6.1	Implication/Scope	16
6.2	Limitations	16
6.3	Conclusion	16
7	References	17

List of Figures

2.1	Architecture	7
5.1	Extraction	15
5.2	Querying	15
5.3	Querying	15

Chapter 1

Introduction

Corporate companies and recruiting agencies receive thousands of resumes daily, due to which a majority of the companies nowadays use ATS to structure and customize the hiring process according to what the company needs. Applicant Tracking System (ATS) is a software instrument developed for HR-managers to support recruitment, screening, sorting candidates, assess and track applicants efficiently based on their skill set, age, salary demand etc. An ATS can offer several capabilities that help organizations find, hire and retain best candidates.

Recruitment software has revolutionized the way Human Resources (HR) staff operates. Prior to the rise of ATS, the manual nature of the process meant that posting job openings in newspapers and accepting hard-copy applications created a host of challenges and limitations. However, using technologies such as email, Excel files, and online application forms has led to a shift towards more digitized management of procedures over time.

The 1st ATS was introduced in the 1970s, these systems were designed to manage job applications manually, and most of the work was done by human recruiters. During the 1980s, ATS began to incorporate more advanced features, such as resume parsing. The rise of the internet in the 90s changed the hiring process significantly.

During this time, Applicant Tracking Systems began to incorporate more advanced algorithms to help recruiters find the best candidates for open positions. The algorithms could use specific criteria such as job experience, educational qualifications and other relevant factors to evaluate each candidate's application and assign them a ranking score. The rise of mobile technology also led to the development of the system.

1.1 Problem Statement

In today's dynamic and competitive job market, organizations receive a vast number of resumes from candidates for all the job openings. Managing all the applications in a fair and efficient way is a tedious task. An applying candidate may be a great fit for the organization but not necessarily in the role they're applying for.

If the resume is screened by a human, then they can easily revert back to the candidate to apply where they are a better fit. However, due to the screening process being automated, the system might reject the application right away, causing the company to lose a great potential employee.

On the other hand, as students who are about to graduate and seek to job-hunt soon, the hiring process seems extremely daunting. Companies nowadays prefer candidates who dabble in multiple domains rather than expertise in a single field. This makes it difficult to navigate the roles available, not knowing what job role is perfect for one's multi-domain skill set. And one can simply get rejected by the system if their resume doesn't match the one role they're applying for.

The motivation behind our model is to serve the purpose of both the employing organization and the candidate applying for roles. The model will scan the text content of the resumes for all the relevant information about a candidate. We will then create a database storing all this information.

The organization can access it as and when required and have a record of all the candidates that apply. This will ensure that even if a candidate is not suited for the position they apply to, the company can still access their information for other job openings.

Chapter 2

Block Diagram

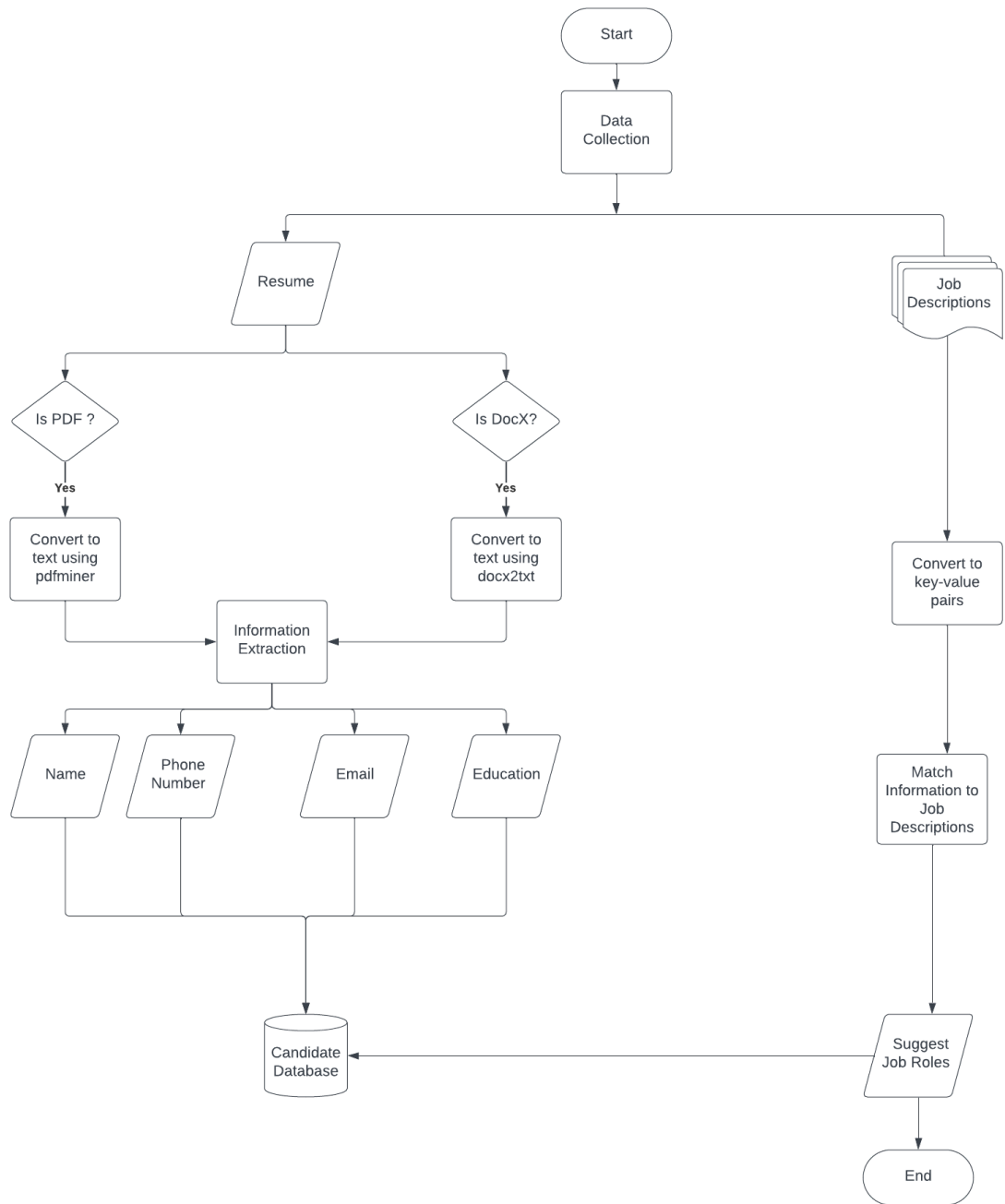


Figure 2.1: Architecture

Chapter 3

Literature Survey

3.1 Review of Papers

[1]

- **Name of Paper:** A Resume Analyzer Application for Matching Candidates With Job Requirements, Using A Parsing Algorithm. .

- **Authors:** Ali, A.A., Olow, F.H., Dahir, A.A., Hassan, A. and Mo-hamud, M., 2021.

- **Summary:** Job matching system is an application which connects employer and job seekers where employers are the source of the resources and the job seeker can find and apply for their targeted job. To extract data from the resume we have used Pyresparser, which is Python module for information extraction from the resumes that has features of extract name, mobile, email, education, skills and experience and more.

- This presents a study in progress work which aims to define the main attributes required for the matching resume data and job offer in the job matching system. This attributes are extracted from job seekers resumes by using resume parser. When the attributes are defined, the proposed framework is guided to follow a word similarity algorithms for the PostgreSQL trigram module to match the job seekers' skills against the requirements of the job posted by the prospect employers.

[2]

- **Name of Paper:** “ATS BREAKER”- A System for Comparing Candidate Resume and Company Requirements *Volume:05/Issue:11*

- **Authors:** Lino Mathew, Nithin C George, Nikitha Linet, Nithin K Thomas (2020) Dept. of Computer Science and Engineering Amal Jyothi College of Engineering Kottayam, Kerala, India

- **Summary:** This paper proposes a model that computes a comparison and based on the results, suggestions are provided to the candidates to modify their resume. The proposed procedure extract information related to technical as well soft skills from the CV’s submitted in text, pdf or docx format. The system also provides suggestions for correcting the grammatical errors. The proposed system is designed based on Natural Language Processing (NLP) techniques. It also successfully stores the analyzed results, which the user can refer to for future purpose.

[3]

- **Name of Paper:** RESUME PARSER USING ML AND NLP *Volume:05/Issue:11*

- **Authors:** Prof. P.R. Kulkarni, Yash Vaidya, Akshad Shelare, Narayan Attarde, Meet Sali (2023)

- **Summary:** This paper introduces a novel Resume Parser and Enhancement System (RPES) designed to revolutionize the way resumes are generated and refined. The proposed system utilizes state-of-the-art natural language processing (NLP) techniques, machine learning algorithms, and recommendation systems to automate the resume building process and provide personalized suggestions for improvement. The core functionality of the RPES involves efficient extraction of information from unstructured resumes, enabling seamless parsing of diverse document formats. Leveraging advanced NLP models, the system ensures accurate identification and categorization of key resume components, such as personal details, education, work experience, skills, and achievements.

[4]

- **Name of Paper:** End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT.

- **Authors:** Vedant Bhatia, Prateek Rawat, Ajit Kuma, Rajiv Ratn Shah (2019)

- **Summary:** First, a resume parser is built which extracts complete information from candidate resumes. This parser is made available to the public in the form of a web application. Second, we use BERT sentence pair classification to perform ranking based on their suitability to the job description. To approximate the job description, the description of past job experiences by a candidate as mentioned in their resume is used.

[5]

- **Name of Paper:** NLP Based Resume Parser Analysis *Vol.10, Issue 5*.

- **Authors:** Mhaske Harshada, A. N. Kshirsagar, Nevase Sonali, Pimparkar Ankita (2023)

- **Summary:** The suggested Employee Recommendation System is utilized to choose individuals from college campuses and other recruitment processes. In essence, candidates for jobs respond to the recruitment staff, who then test them to see if they qualify for a particular designation. This is a laborious profession that necessitates much interpersonal contact during the hiring process.

- It might also involve various hiring team must also check the candidates' backgrounds to see if they have engaged in any strange behaviors, such having a criminal record. Recruiting agencies and business enterprises review a number of resumes daily. The name, email address, social media profiles, personal websites, years of work experience, artwork examples, years of education, coaching studies, publications, certifications, volunteer experiences, key phrases, and sooner or later the cluster of the resume are all included in the parsed statistics. A database is then used to store the parsed files for subsequent usage.

[6]

- **Name of Paper:** Resume Parser with Natural Language Processing.

- **Authors:** Pornphat Sroison, Jonathan H. Chan: December 29, 2021

- **Summary:** Parse resume and match resume to job description are the two functions of this system. The first function is to parse resumes. The user must upload a resume of the candidate file in PDF or DOC format. The system will read all text of the resume and extract only relevant data that is necessary for the selection of the resume: name (first name and last name), position applied for, university, degree, work experience, skill, email address and phone number. The second function is matching resumes to job descriptions to evaluate how similar they are. The user can upload a job description file and see the displayed result as a percentage of similarity between resume of candidate and job description.

- The proposed system's results are shown in this part, which include extracting name, designation, university, degree, skills, experience, email and phone number using Named Entity Recognition to develop a model and Regular Expression to extract the data. It is also possible to compare the applicant's resume and the job description to see the percentage of similarity as well. This system can assist the human resource department or employer in screening resumes before conducting interviews and finding the best candidate for the job position.

Chapter 4

Methodology

4.1 Research Design

We studied various papers on the usage of ATS, systems to compare candidates to company requirements, usage of Machine Learning, Natural Language Processing and recommendation systems.

The data collected was of 2 types. Firstly, we circulated a google form to collect resume data. The target demographic were people studying in the technical field. Resumes were in 2 formats (PDF and DOCX) and the text content was mined from both using different python modules. Secondly, we used an open-source website (workable.com) that is curated as job role templates for hiring managers to get data on several job descriptions in the data-oriented and technical fields.

4.2 Data Analysis

i) Extraction of Information from text: Regular Expression or RegEx in Python are a powerful tool for pattern matching and text manipulation. They allow to search for specific patterns within strings and perform various operations like finding, replacing, or extracting text based on those patterns. We can define string patterns specific to the kind of information we want to save (name, contact information, education, experience and skills) and extract relevant texts easily.

ii) Matching resumes to job roles: The Natural Language Toolkit (NLTK) is a Python package for natural language processing. Punkt Sentence Tokenizer divides a text into a list of sentences by using an unsupervised algorithm to build a model for abbreviation words, collocations, and words that start sentences. `word_tokenize` splits a given sentence into words using the NLTK library.

A stop word is a commonly used word (such as “the”, “a”, “an”, or “in”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query.

A corpus can be defined as a collection of text documents. By eliminating these words from a corpus, we can more easily identify unique and relevant terms.

Formally, a frequency distribution can be defined as a function mapping from each sample to the number of times that sample occurred as an outcome. The `FreqDist` class is used to encode “frequency distributions”, which count the number of times that each outcome of an experiment occurs `sklearn.feature_extraction.text.TfidfTransformer` - count matrix to a normalized `tf` or `tf-idf` representation. `Tf` means term-frequency while `tf-idf` means term-frequency times inverse document-frequency.

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space based on the cosine of the angle between them, resulting in a value between -1 and 1. The value -1 means that the vectors are opposite, 0 represents orthogonal vectors, and value 1 signifies similar vectors. The cosine similarity is often used in text analytics to compare documents and determine if they’re similar and how much.

iii) Database: SQL magic command in Jupyter notebooks is a way to interact with databases directly within a Jupyter environment. By using `%%sql` at the beginning of a cell in a Jupyter notebook, you can write and execute SQL queries as if you were working in a database client. For instance, if you have a database connection set up in your Jupyter notebook, you can use `%%sql` followed by your SQL query to fetch data, perform operations, or manipulate the database. You can use different SQL dialects depending on the database you’re working with.

4.3 Ethical Considerations

Everyone who submitted their resume in the form was informed of the purpose of data collection and has consented to use their data to be worked on in this project. It will be kept strictly confidential aside from the people involved in implementing and grading the project. The data on job descriptions for all the roles was taken from an open-source websites that serves templates for hiring managers to curate roles for their companies. The website is free for everyone to use as reference.

Chapter 5

Results and Discussions

- Accurately extracted relevant information from resumes such as name, contact information, education, CGPA, required skills for a specific job role.
- Identification of key skills, experiences, and qualifications that match specified job requirement (role of Frontend Engineer in this case).
- Stored the information of candidates in database for future reference.
- Matching candidate with relevant job profiles based on their skills, experience, education and other factors.
- Sort candidates for different roles earlier in the hiring process.
- Get list of candidates as per requirement. (CGPA = 8) and (Role == Analyst)
- Delivering fast and reliable recommendations.
- Handles variations in resume formats efficiently.
- Minimization of errors in information extraction.

```
In [15]: #basic information

print("Name:", extract_name_from_resume(s1))
print("Contact Number:", extract_contact_number_from_resume(s1))
print("Email:", extract_email_from_resume(s1))
print("Education:", extract_education_from_resume(s1))
print("CGPA:", extract_gpa_from_resume(s1))

#skills for a specific role
skills_list = ['frontend','HTML', 'HTML5', 'CSS', 'CSS3', 'React','ReactJS','Angular','AngularJS','jQuery','Frameworks','Version']
print("Skills that match specified role:", extract_skills_from_resume(s1, skills_list))

#roles that match candidate skills
print("Top roles that match candidate skills:", calculate_keyword_match(s1, jobRoles))
```

Name: Kirti Patel
Contact Number: 8655204810
Email: kirtiipatel4@gmail.com
Education: ['Bachelor of Technology']
CGPA: 9.43
Skills that match specified role: ['React', 'Optimization', 'Testing', 'Debugging', 'Web Development']
Top roles that match candidate skills: ['Frontend Engineer', 'Data Analyst', 'Blockchain Engineer']

Figure 5.1: Extraction

```
In [34]: %%sql

SELECT * FROM CANDIDATES WHERE CGPA > 8.0;

* sqlite:///SQLiteMagic.db
Done.
```

Out[34]:

Name	Contact_Number	Email	Education	CGPA	Role1	Role2	Role3
Kirti Patel	8655204810	kirtiipatel4@gmail.com	Bachelor of Technology	9.43	Frontend Engineer	Data Analyst	Blockchain Engineer
Vansh Mehra	9711982402	vansh_m@iitr.ac.in	B.Tech	8.17	Data Analyst	Data Engineer	Data Scientist
Zenia Noupal	9653392978	zenianoupal5@gmail.com	B.Tech	9.43	Data Analyst	Business Analyst	Data Engineer
Mrunal Chavan	8425808902	mrunalchavan1973@gmail.com	B.Tech	8.9	Data Analyst	Machine Learning Engineer	Data Engineer
Sahil Raj	8454800513	raj17sahil@gmail.com	Bachelor of Engineering	8.73	Data Scientist	Blockchain Engineer	Data Analyst
Nikita Shetty	9867332365	shettynikita926@gmail.com	Bachelor of Technology	8.71	Backend Engineer	Frontend Engineer	Database Administrator

Figure 5.2: Querying

```
In [35]: %%sql

SELECT * FROM CANDIDATES WHERE Role1 == 'Data Analyst';

* sqlite:///SQLiteMagic.db
Done.
```

Out[35]:

Name	Contact_Number	Email	Education	CGPA	Role1	Role2	Role3
Vansh Mehra	9711982402	vansh_m@iitr.ac.in	B.Tech	8.17	Data Analyst	Data Engineer	Data Scientist
Zenia Noupal	9653392978	zenianoupal5@gmail.com	B.Tech	9.43	Data Analyst	Business Analyst	Data Engineer
Mrunal Chavan	8425808902	mrunalchavan1973@gmail.com	B.Tech	8.9	Data Analyst	Machine Learning Engineer	Data Engineer
Janhavi Dixit	9370680015	janhavidixit249@gmail.com	Bachelor of Technology	0.0	Data Analyst	Backend Engineer	Database Administrator
Sejal Londhe	7715084590	sejallondhe14@gmail.com	Null	7.66	Data Analyst	Backend Engineer	Business Analyst
Purva Patil	9620538301	purvap0078@gmail.com	Null	0.0	Data Analyst	Data Scientist	Machine Learning Engineer
Nidhi Ravariya	9892700632	nidhiravariya256@gmail.com	B.Tech	7.51	Data Analyst	Business Analyst	Machine Learning Engineer

Figure 5.3: Querying

Chapter 6

Conclusion

6.1 Implication/Scope

The model can be further developed or optimized using semantic understanding which means analyzing the context used behind each word instead of simply matching them. Multilingual NLP capabilities may be integrated to serve a larger demographic, aiding the globalization of job markets. Advancement in AI/ML algorithms can greatly improve the accuracy of these systems.

The model can be built into an application, a platform available to both candidates and employers / hiring managers. Candidates can have an option to upload their application and get a list of suitable roles and openings available. Employers can post an open role and collect information of candidates that apply for the same.

6.2 Limitations

Had to manually convert job description text to key value pairs as there's no way to accurately capture domain related keywords. Also had to manually store the information in the database instead of the function output being stored in the database automatically.

6.3 Conclusion

This solution addresses the drawback that arises as a result of automated resume screening. By using natural language processing techniques, the model intends to analyze resumes submitted in text document format to accurately assess how well a candidate profile fits the job requirement. The system will simultaneously store the data and scan other job descriptions to seek a better match. Ultimately, it will further optimize the recruitment process to better adapt to today's dynamic job market.

Chapter 7

References

- [1] Ali, A.A., Olow, F.H., Dahir, A.A., Hassan, A. and Mohamud, M., 2021. A Resume Analyzer Application for Matching Candidates With Job Requirements, Using A Parsing Algorithm.
- [2] Lino Mathew, Nithin C George, Nikitha Linet, Nithin K Thomas (2020) Dept. of Computer Science and Engineering Amal Jyothi College of Engineering Kottayam, Kerala, India: “ATS BREAKER”- A System for Comparing Candidate Resume and Company Requirements *Vol. 9 Issue 06*
- [3] Prof. P.R. Kulkarni, Yash Vaidya, Akshad Shelare, Narayan Attarde, Meet Sali (2023) RESUME PARSER USING ML AND NLP , *Volume:05/Issue:11*
- [4] Vedant Bhatia, Prateek Rawat, Ajit Kuma, Rajiv Ratn Shah (2019) End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT.
- [5] Mhaske Harshada, A. N. Kshirsagar, Nevase Sonali, Pimparkar Ankita (2023) NLP Based Resume Parser Analysis Vol.10, Issue 5.
- [6] Pornphat Sroison, Jonathan H. Chan: Resume Parser with Natural Language Processing. December 29, 2021