



---

## **Recommendation for Jobs and Resume Analyzer Using NLP**

**Jasmit Gharat**

MCA Department, Sardar Patel Institute of Technology  
[jasmit.gharat@spit.ac.in](mailto:jasmit.gharat@spit.ac.in)

---

### **ABSTRACT**

*Nowadays companies use ATS to scan resumes of the applying candidates. Thousands or even lakhs of candidates apply to the same position every time. With the help of ATS the company shortlists some candidates based on their relevant skills. Now the HR or the Recruiting team has to manually look at these resumes. This can be a tedious job for an HR after all he or she is also a human being. This system can be used to overcome this tedious task. Hr can just upload the resumes and this will extract all the relevant skills or data of the candidate and after which the HR can choose the best candidates suitable for the position. This will save a lot of time of the recruiting Team and can seriously eliminate the additional headache of reviewing the resumes manually thereby increasing the efficiency of the company.*

---

### **I. INTRODUCTION**

The majority of the resume screening is done using Natural Language Processing (NLP), which relates to how people communicate with one another. Giving computers the ability to comprehend spoken and written language in a manner similar to that of humans is the goal of NLP. Statistical, machine learning, and deep learning models are all combined in natural language processing. Millions of new workers enter the workforce each year as the employment market in India is expanding. According to the Employees Provident Fund Organisation 1.3 million new positions will be generated by 2022. In order to hire the appropriate candidate at the right time recruiters must be able to thoroughly review resumes. Resume screening is the process of evaluating a candidate's suitability for a position based on factors such as credentials, education, work history, and other data from their CV. In this article, we propose a resume screening method that uses NLP to create a candidate profile that matches the necessary competencies and is plotted as a bar chart for improved visualisation.

---

### **II. OBJECTIVE:**

*To fetch or extract relevant data from resumes. It is important to extract data such as Education, Skills, Experience, Achievements, Projects. Help HR to easily compare resumes between multiple candidates and select the best suite for the company. Provide the best recommendation based on the user's skills. To provide an admin module for the admin or HR to analyse all the resumes that are uploaded over the portal. To Provide Relevant real Job recommendation to users based on their resume. Users can search for jobs based on the recommendation that the user will get upon his resume. HR can use the admin module for data visualisation purposes using various statistical methods such as graph, charts, bars etc.*

---

### **III. SCOPE**

*It may be used to convert all of the resume data into structured tabular and csv formats so that the organisations can use them for analytics. Users can improve their resume and can use the system for advice, projections, and total score of their resume. Furthermore the user section can improve traffic to our tool. Prior to placements institutions might use it to gain insight into students' resumes moreover to obtain analytics for the roles that people are primarily searching for. Gaining feedback will help to enhance this tool. This system can be useful for the HR department to hire the best candidates and can also be used for their daily work review with the help of pie-charts for visualisation.*

---

### **IV. RELATED WORK**

[A] *Resume Analyzer Using Text Processing:*

(B. Kelkar, R. Shedbale, D. Khade 2)

In order to assist recruiters in choosing the best applicant for a particular position, this literal review offers a powerful Company Recommender System that makes use of text mining and machine learning algorithms. Candidates are graded based on the specifications of the company after uploading their resumes. The company might utilise the rating to choose the top applicants. The approach and model for this article will be presented in four steps: gathering resumes and looking up keywords in the database that contains data about the text of the resumes. Next, applicants were ranked and categorised based on their rating scores. For the purpose of enhancing the knowledge base, this system may also extract fresh keywords from resumes.

**[B] NLP Based Extraction of Relevant Resume using Machine Learning:**

(Nirali Bhaliya, Jay Gandhi, Dheeraj Kumar Singh 2)

According to this method, resumes are parsed with the fewest possible characters, and the parser uses two or three guidelines that teach call and address. The system of the CV parser is used by Scout bundles to determine resumes. Due to the incredible arrangements of resumes and the various types of important ingredients they have, like setup and estimations made without a plan, meta-experiences, etc. The parsing of this ingredients from the moved CV's is provided by the proposed CV parser methodology.

**[C] E-Recruitment System through Resume Parsing, Psychometric Test and Social Media Analysis:**

(Dr Parka Department of Computer Science and Engineering MS Ramaiah Institute of Technology Bangalore 2)

The process is divided into four steps. The first stage involved gathering the data that is an applicant resume and structuring it before performing the analysis using deep learning methods. The psychometric test is the second step, and results for each candidate are generated via text mining. In the third phase, they use web scraping to gather further information about individuals and suggest jobs that would be a good fit for them. In the fourth step, the system will make recommendations for the abilities and prerequisites that the students need to improve upon and will assist them in getting hired by the targeted organisation.

**[D] CV Parser Model using Entity Extraction Process and Big Data Tools:**

(Das, Papiya & Pandey, Manjusha & Rautaray, Siddharth. (2018). International Journal of Information Technology and Computer Science. 2)

Here, the specification of the challenge was focused on creating an automatic resume parser system that would analyse the uploaded resume in accordance with the job profile. Additionally, it will change the unstructured resumes into a manner that is structured. Additionally, a ranking system for resumes will be maintained. Ranking will be determined by the information retrieved, such as technical expertise, level of education, etc. Here, a CV parser is employed. One method for gathering CVs is CV parsing.

## V. ALGORITHMS:

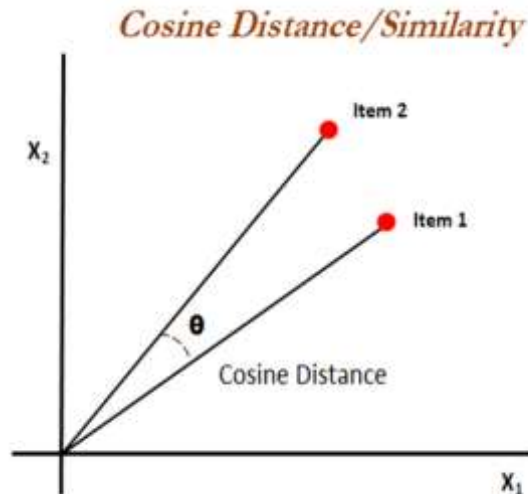
**[A] Cosine Similarity**

`import cosine_similarity`

*Calculate the cosine similarity between courses.*

*The cosine similarity between two vector pairs is calculated using this function.*

*The result is a similarity matrix that contains the pairwise cosine similarities between all pairs of vectors in vectors. When two vectors have a cosine similarity of 1, they are said to be identical, and when they have a cosine similarity of 0, they are either orthogonal or utterly different. The cosine similarity values vary from -1 to 1, where -1 denotes that the two vectors are orthogonal, 0 that they are opposite in direction, and 1 that they are identical.*



**Fig 1:** Cosine Similarity between two courses

### [B] Recommendation Function

Create a function that will iterate through the courses and find other courses with the help of course\_index of similarity.

The index of the input course is first retrieved in the new\_df DataFrame.

The initial index of the generated DataFrame is then taken after selecting rows where the course\_name column equals the supplied course.

The row in the similarity matrix corresponding to the input course is then retrieved. The cosine\_similarity function was used to create the similarity matrix.

A list of tuples representing the cosine similarity between the input course and each other course in new\_df is then created. A list of tuples containing the index of each course in new\_df and its cosine similarity to the input course are first created using the enumerate function. The top 6 courses (other than the input course itself) are then chosen after this list is sorted in descending order based on the second element of each tuple (i.e., the cosine similarity).

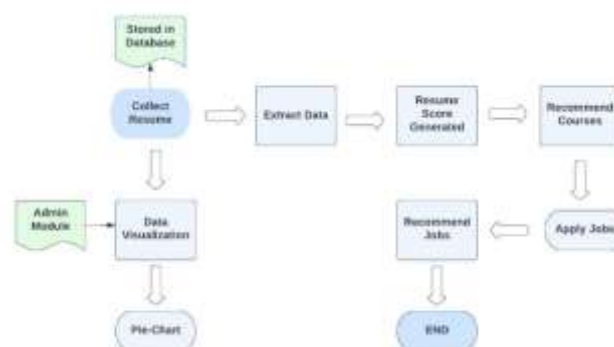
Finally, it loops through the recommended courses in the course\_list and returns their names.

### [C] ResumeScore(NLP):

Based on the content that a resume has in it, a score is generated out of 100 marks. There are five factors or parameters that analyse the resume and generate a score for that resume out of 100. Since NLP is used to extract text data from resume, now after successful extraction of data there is a function created to check whether the resume contains certain content or not. These contents are hobbies, career objective, declarations, achievements and projects. If either of the content is absent in the resume then it will deduct 20 marks and thus final marks will be calculated based on the algorithm.

## VI. METHODOLOGY:

The following methodology will be followed for the development of the Resume Extractor:



**Fig 2:** WorkFlow of entire system

**[A] Model Creation:**

The content-based similarity filtering is used in this course recommendation system. The coursera courses dataset which has approximately 3000 courses is the dataset that was used.

**[1] Basic Data Analysis**

Importing Dependencies and reading CSV file and knowing the structure of data by using `data.shape()` `data.info()`.

Look for null values, count the number of institutions offering courses, the number of courses in each difficulty range and the rating values.

Important columns to be used are Course Name, Course Description, Skills and Difficulty Level. Whereas columns not used are Course Ratings because distribution is not even, Course URL as it is of no significance and University because Same university might offer multiple courses in different domains which the user might not want to see

```
In [6]: data['Difficulty Level'].value_counts()

Out[6]: Beginner          1444
         Advanced          1005
         Intermediate       837
         Conversant         186
         Not Calibrated      50
```

*Fig 3: Courses Category*

**[2] Data Pre-Processing**

Pre-processing the data into a form that can be used by the recommendation system is a crucial step in the procedure. This also involves cleaning the data such as removing additional spaces between the words and also removing unwanted parenthesis from skills columns

**[3] Dataframe to be used**

The tags column is made by merging Course Name, Difficulty Level, Course Description, Skills. Now create a different dataset with only required columns after preprocessing data.

**[4] Text Vectorization**

Create a vector for each course. Remove the stopwords such as a, and, the etc. Text Vectorization is done to calculate the cosine similarity.

**[5] Stemming Process**

```
import PorterStemmer from nltk.stem.porter
```

create the stemming function

stemming is applied on the tags column

**[6] Exporting the Model**

```
import pickle
```

use pickle to import the model into streamlit web app.

object is being saved to a file named 'similarity.pkl' using the "pickle.dump()" function.

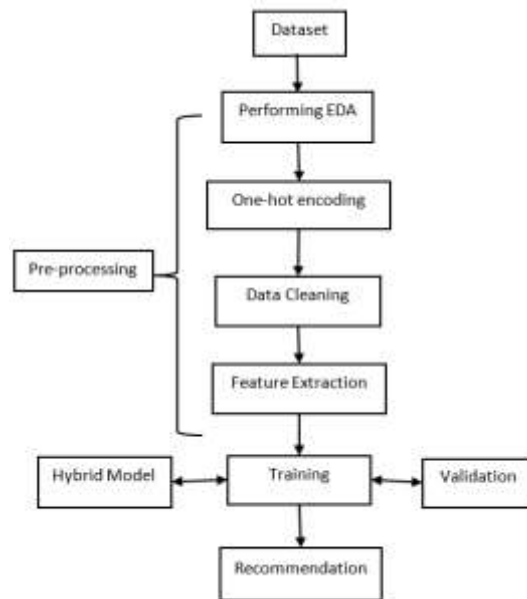


Fig 4: Flow Chart for Recommendation System

**[B] Admin Module(database connectivity):**

Mysql has been used to store the candidate data. In sys database a new table is created(if not already created) with columns(schema) that are name, email, resume score, timestamp, no of pages, recommendation field, experience level and skills. As the user uploads the resume all the above details are extracted from the resume and are stored in the database with the above schema. Further this data is used for further analysis and visualisation(pie-charts).

**[C] Pyresparser:**

Import ResumeParser from the Pyresparser library. It is an open source library that will take the location of the resume(pdf or doc) as input and it will extract content from the resume in json format. The content that it will extract is college name, company names, degree, designation, email, mobile number, name, no of pages, skills and total experience. Used the skills that have been extracted for further analysis.

**[D] Job Recommendation:**

<https://rapidapi.com/fastapply/api/job-board-search-fastapply>

The above API that is available on the rapid API has been used to get the real time job details across the globe. This API fetches data from different job portals and displays the result. Users have to enter role and location. The API will fetch jobs according to the data provided by the user and it will display the results along with other details such as jobID,min and max salary range and other job details.

## VII. RESULTS:

[A] Course recommender where the user selects a course according to his preference and gets recommendation of courses from coursera.



Fig 5: Course Recommender

[B] Users choose location and role to search for jobs and get real time job data.

### Apply For Jobs

Enter job title:  
WebDeveloper

Enter location:  
Mumbai

	location	job_title	publisher_name	publisher_link
0	Mumbai, Maharashtra, India	Web Developer	Indeed	<a href="https://in.indeed.com/cmy">https://in.indeed.com/cmy</a>
1	Mumbai, Maharashtra, India	Full Stack Web Developer	Glassdoor	<a href="https://www.glassdoor.com">https://www.glassdoor.com</a>
2	Mumbai, Maharashtra, India	Web Developer	Payscale	<a href="https://www.payscale.com">https://www.payscale.com</a>

Fig 6: Job Application

[C] Admin module where admin can check data of all the users.

### User's Data

ID	Name	Email	Resume Size	Timestamp
0	21_0120New York	info@resumekraft.com+1-202-555-0120New	60	2023-04-04_17:1
1	34 Android Developer	info@resumekraft.com	40	2023-04-04_13:0
2	19_0120New York	info@resumekraft.com+1-202-555-0120New	60	2023-04-04_13:0
3	17 Chiranjay Khagwal	chiranjay.khagwal@prn.ac.in	40	2023-04-04_18:5
4	82 Jaanil Ghatak	jaanilghatak100@gmail.com	20	2023-04-04_09:2
5	83 Jaanil Ghatak	jaanilghatak100@gmail.com	20	2023-04-04_09:2

Fig 7: Admin Section- Users Data

[D] Pie-chart visualisation of all users data based on experience level

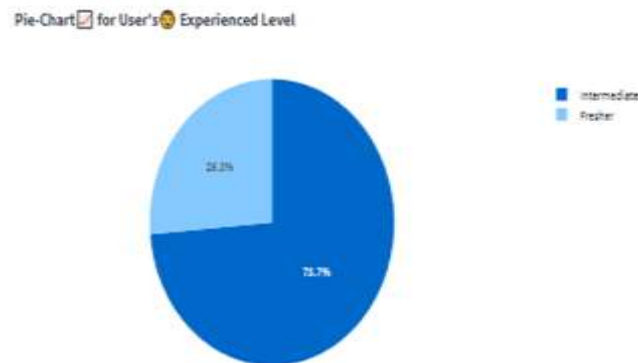


Fig 8: Pie Chart- Experience Visualizer

## VIII. CONCLUSION:

Future research will focus on other libraries, new means of exploring various strategies, and a more in-depth analysis of specific techniques. The existing datasets are insufficient for applications like designation, university, skill, etc., so this project aims to supply more datasets for training in the future. Ranking of resumes can be done based on the job description. Extraction of resumes from various applications and websites, like LinkedIn, GitHub, Naukri.com, and others can be done in the future.

In order to shorten the time it takes for an organisation to hire a candidate, the resume parsing technique seeks to be utilised for applicant profile screening. In order to make the hiring process for applicants easier for recruiters, CV parsing involves showing a candidate's skill set in the form of a bar graph. The system can successfully convert many resume formats to text format and retrieve certain key information. The proportion of similarity between the applicant's resume and the job description can also be determined by comparing the two. This approach can help the human resources department to review resumes prior to conducting interviews and choosing the best candidate for the job. The system was able to convert all the resumes to text and extract crucial information from there. The model aims to make employers and candidates' jobs easier and more efficient. Real time applying for Job becomes much easier with the extra job application feature that has been added in the system thus enhancing user experience.

**IX. REFERENCES:**

- [1] B Kelkar, R Shedbale, D Khade, Professor, Dept. of Computer Science and Engineering, Sanjay Ghodawat College, Shivaji University, India, "Resume Analyzer Using Text Processing", Vol 11, Issue 5, May/2020 ISSN NO:0377-9254
- [2] Nirali Bhaliya, Jay Gandhi, Dheeraj Kumar Singh, "NLP Based Extraction of Relevant Resume using Machine Learning", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075 (Online), Volume-9 Issue-7, May 2020
- [3] Dr Parka A parkavia@m.snt.edu, Department of Computer Science and Engineering MS Ramaiah Institute of Technology Bangalore. "E-Recruitment System through Resume Parsing", Psychometric Test and Social Media Analysis: Vol.5, Issue.7, July 2019
- [4] Das, Papiya & Pandey, Manjusha & Rautaray, Siddharth(2018). International Journal of Information Technology and Computer Science. "CV Parser Model using Entity Extraction Process and Big Data Tools". 10.5815/ijitcs.2018.09.03.
- [5] Courses Dataset 2021(Coursera) <https://www.kaggle.com/datasets/khusheekapoor/coursera-courses-dataset-2021>
- [6] Pyresparser Library By Omkar Pathak <https://omkarpathak.in/pyresparser/>
- [7] Recommendation Systems in Machine Learning <https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning>
- [8] Building Recommender system with Python <https://www.analyticsvidhya.com/blog/2021/08/developing-a-course-recommender-system-using-python/>
- [9] Zu, Shicheng, Wang, Xiulai, Darren, Seth "Resume Information Extraction with A Novel Text Block Segmentation Algorithm". Linguistics 10.5121/ijnlc.2019.8503
- [10] International Journal of It and Computer Science 10(9):21-31 DOI:10.5815/ijitcs.2018.09.03  
[https://www.researchgate.net/publication/327711084\\_A\\_CV\\_Parser\\_Model\\_using\\_Entity\\_Extraction\\_Process\\_and\\_Big\\_Data\\_Tools](https://www.researchgate.net/publication/327711084_A_CV_Parser_Model_using_Entity_Extraction_Process_and_Big_Data_Tools)
- [11] Recommender system with Python <https://www.analyticsvidhya.com/blog/2021/08/developing-a-course-recommender-system-using-python/>
- [12] Abstractive vs Extractive Text Summarization (Output based approach) - A Comparative Study <https://ieeexplore.ieee.org/abstract/document/9298416>
- [13] AutoKYC: Automation of Identity establishment and authentication in KYC process using Text extraction and face recognition <https://ieeexplore.ieee.org/abstract/document/9909442>
- [14] Offline Voice Commerce Shopper Tracker Using NLP Analytics [https://link.springer.com/chapter/10.1007/978-981-33-6546-9\\_35](https://link.springer.com/chapter/10.1007/978-981-33-6546-9_35)