An Automated Recommendation System for Job Search Analysis Using Stochastic Gradient Descent Model with Enhanced Keyword Search Platform

Akshaya Motamarri
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India

Arunima Agarwal
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India

A. Sheik Abdullah
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India

***Abstract*—Due to the increase in population, the overall growth of a country reduces. To stabilize this growth, increasing the employment rate is one of the major solutions. With many candidates applying for jobs, recruiters spend a lot of time manually reviewing resumes or curriculum vitae. Searching through the resumes of thousands of job applicants has become a challenging task. To overcome this difficulty, recent research has focused on building machine-learning models that can be adapted to different resume styles and formats. This work proposes a model that can create resumes for each candidate based on the information they provide, helping recruiters intelligently sift through thousands of resumes so they can select the right candidate for a job interview using the resume extraction technique. Resume extracting or parsing is used to find structured information from unstructured data in the resume, to facilitate its storage, analysis, and management. It is a very useful tool for recruiters, and human resource professionals, as it allows them to quickly and efficiently identify key information about job candidates and make it easier to shortlist the candidates.**

***Keywords*—Curriculum Vitae, NLP, Python,R**

I. Introduction

Daily, commercial enterprises and retaining agencies have to reuse a large number of resumes. Working with a large volume of text data is generally TIME-CONSUMING and stressful. Data gathered from different resumes can be in various FORMS, including .pdf, .docx, SINGLE-COLUMN resumes, DOUBLE-COLUMN resumes, free formats, and so on. And these formats might not be suitable for the particular application. This is where the concept of resume parsing comes into the picture. The process of converting the unstructured form (.pdf/.docx/.JPEG ETC.) of resume data into a structured format is known as resume parsing. latterly, converting a resume into prepared text or structured information makes studying, assaying, and comprehending easier.

As a result, numerous associations and institutions depend on Information Extraction, where unstructured data and vital information are uprooted and converted to make information more readable and systematized data forms. The completion of this task takes a long time for humans. Therefore, it's necessary to develop an automated intelligent system that can pull all applicable information to determine whether an aspirant is suitable for a particular job profile. Each needs to analyze their skills before going for an interview, this is because only then they will be able to answer the interviewer confidently.

This model will be built using natural language processing (NLP) methods to extract pertinent information from resume text. Then, using a smaller, annotated dataset of resumes, where the pertinent data has been tagged by humans, this model is refined. This enables the model to pick up on more precise patterns and traits that point to crucial resume information. The system comprises the use of the Python spaCy library.

Also, we will be building an interface using streamlit package in Python3 which allows us to create a front-end and back-end. We then connect the MySQL database to Python by using a package called pymysql. We will be using packages like pdf miner and geocoder for an interactive interface.

Also using R programming, we will be analyzing the uploaded resume file and tokenizing each attribute to get statistical data from the file such as word count, and word frequency, and make plots for those frequency distributions. We will incorporate word cloud package to check easily. The packages used will be tidyverse, tm, readtext, tidytext, ggplot2, RColorBrewer, wordcloud, and syuzhet. These packages help in lemmatizing the given file in text format.

II. RELATED WORK

The ever-adding number of operations to job positions presents a challenge for employers to find suitable campaigners manually. BERT was used for the sequence brace bracket to rank campaigners as per their felicity to a particular job description. With the data collected from the web operation, we will have accurate job descriptions and canvasser commentary at each stage of the hiring process. Also, it's planned to further explore the vision-grounded runner segmentation approach to compound our structural understanding of resumes. This work establishes a vital birth and evidence of conception which can lead to the hiring process serving from the advances in deep literacy and language representation.

The designed and enforced curriculum vitae parsing system combines wordbook styles with styles for named reality recognition. Three NER tools (Liner2, NERF, Babelfy), the anchor system, and different types of wordbooks were used. This mongrel approach allowed for significant content of information birth. The anomaly discovery in textbooks and handling outliers in resumes are still gruelling tasks. The prototype doesn't include the outliers in the posterior phases of the analysis. still, applying insulation timber or FuzzyGrounded insulation timber may be salutary for the reclamation process. A fresh and promising CRF system was also applied to the proposed system. Although the original tests didn't bring the anticipated results, they revealed that outlier discovery might be salutary in CV document analysis. Thus, further examinations should concentrate on this sphere. Named reality recognition (NER) is still an open problem, especially for low-resource languages. The addition of richer verbal information (e.g., reliance parsing) and adaption of the current systems are intriguing avenues to explore.

The estimations set of immense records are enormous and jumbled. Consequently, colourful item programs have been added to deal with similar considerable databases. CV parsing is such a strategy for a social occasion. CV parser reinforces further than one language, Semantic mapping for limits, development wastes, determination agents, and effortlessness of customization. Parsing with parcel limit bears us accu-cost results. Its age accelerates for mentioning resumes concerning to its feathers and codecs. Its collaboration advances guests' API keys for mixed endeavors. The parser works with the application of two or three rules which train the call and address. Scout packets use the CV parser system for the determination of resumes. As resumes are in amazing arrangements, they have different feathers of real factors like set-up and unshaped estimations, meta gests, etc. The proposed CV parser approach gives the element birth system from the moved CVs.

While the Internet takes up by far the most significant part of our diurnal lives, finding jobs/ workers on the Internet has started to play a pivotal part for job campaigners and employers. Online reclamation websites and mortal coffers consultancy and reclamation companies enable job campaigners to produce their résumé, a brief written formal document including job candidate’s introductory information similar to particular information, educational information, work experience, and qualifications to find and apply for desirable jobs. In contrast, they enable companies to find good workers they're looking for. Still, résumés may be written in numerous ways that make it delicate for online reclamation companies to keep these data in their relational databases. In this study, a design that Kariyer.net (the largest online reclamation website in Turkey) and TUBITAK (The Scientific and Technological Research Council of Turkey) have been concertedly working on is proposed. In this design, a system enables the accessible structured format of résumés to transfigure into an ontological structure model. The produced system grounded on an ontological structure model and called Ontology grounded Résumé Parser (ORP) will be tested on several Turkish and English résumés. The proposed system will be kept in the Semantic Web approach that provides companies to find expert findings effectively[1].

Today, the proportion of bits of knowledge timber is incredibly tremendous. Dependent upon the adaptations of estimations, immense information involves social, machine, and trade-based Data. Social estimations gathered from Facebook, Twitter, etc. Machine information is RFID chip examination, GPRS, etc. Trade-based bits of knowledge consolidate retail point information. Around the assortments of different feathers of estimations first member is published happy real factors. Content information is sorted out information. Inferring high five stars sorted out records from the unshaped published content is a cultural substance examination. Changing over unshaped real factors into critical records is a book assessment process. CV parsing is one of the substance examination strategies. It keeps parsing or extracting of CV.CV parser combines the seeker's capsule with selection gems flow and this way systems move toward CVs. This paper proposes a CV parser adaptation of the operation of cultural substance examination. The proposed CV parser interpretation isolates substances needed in the investiture methodology inside the associations[2].

In the online job reclamation sphere, accurate brackets of jobs and resumes to occupation orders are important for matching job campaigners with applicable jobs. An illustration of such a job title bracket system is an automatic textbook document bracket system that utilizes machine literacy. Machine literacy-based document bracket ways for images, textbooks, and related realities have been well-delved in academia and have also been successfully applied in numerous artificial settings. In this paper, we present Carotene, a machine literacy-based semi-supervised job title bracket system that's presently in the product at CareerBuilder. Carotene leverages a varied collection of bracket and clustering tools and ways to attack the challenges of designing a scalable bracket system for a large taxonomy of job orders. It encompasses these ways in a waterfall classifier armature. We first present the armature of Carotene, which consists of a two-stage coarse and fine-position classifier waterfall. We compare Carotene to an early interpretation that was grounded on a flat classifier armature and also compare and discrepancy Carotene with a third-party occupation bracket system. The paper concludes by presenting experimental results on real-world artificial data using both machine literacy criteria and factual stoner experience checks[3].

Currently, exploration in textbook mining has come one of the wide fields in assaying natural language documents. The present study demonstrates a comprehensive overview of textbook mining and its current exploration status. As indicated in the literature, there's a limitation in addressing Information birth from exploration papers using Data Mining ways. The community between them helps to discover different intriguing textbook patterns in the recaptured papers. In our study, we collected and textually anatomized through colourful textbook mining ways, three hundred refereed journal papers in the field of mobile literacy from six scientific databases, vicelike Springer, Wiley, Science Direct, SAGE, IEEE, and Cambridge[4]. The selection of the collected papers was grounded on the criteria that all these papers should incorporate mobile literacy as the main element in the advanced education environment. Experimental results indicated that the Springer database represents the main source for exploration papers in the field of mobile education for the medical sphere. Also, results, where the similarity among motifs couldn't be detected, were due to either their interrelations or nebulosity in their meaning. Likewise, findings showed that there was a booming increase in the number of published papers during the times 2015 through 2016. In addition, other counter-accusations and unborn perspectives are presented in the study[5].

Currently, exploration in textbook mining has come one of the wide fields in assaying natural language documents. The present study demonstrates a comprehensive overview of textbook mining and its current exploration status. As indicated in the literature, there's a limitation in addressing Information birth from exploration papers using Data Mining ways. The community between them helps to discover different intriguing textbook patterns in the recaptured papers. In our study, we collected and textually anatomized through colourful textbook mining ways, three hundred refereed journal papers in the field of mobile literacy from six scientific databases, vicelike Springer, Wiley, Science Direct, SAGE, IEEE, and Cambridge. The selection of the collected papers was grounded on the criteria that all these papers should incorporate mobile literacy as the main element in the advanced education environment. Experimental results indicated that the Springer database represents the main source for exploration papers in the field of mobile education for the medical sphere. Also, results, where the similarity among motifs couldn't be detected, were due to either their interrelations or nebulosity in their meaning. Likewise, findings showed that there was a booming increase in the number of published papers during the times 2015 through 2016. In addition, other counter-accusations and unborn perspectives are presented in the study[6].

Understanding short texts are pivotal to numerous operations, but challenges abound. First, short texts don't always observe the syntax of a written language. As a result, traditional natural language processing tools, ranging from part-of-speech trailing to reliance parsing, cannot be fluently applied. Second, short texts generally don't contain sufficient statistical signals to support numerous state-of-the-art approaches for text mining similar to content modelling. Third, short texts are more nebulous and noisy and are generated in an enormous volume, which further increases the difficulty to handle them. We argue that semantic knowledge is needed to understand short texts. In this work, we make a prototype system for short text understanding which exploits semantic knowledge handed by a well-known knowledgebase and automatically gathered from a web corpus. Our knowledge-ferocious approaches disrupt traditional styles for tasks similar to text segmentation, part-of-speech trailing, and conception labelling, in the sense that we concentrate on semantics in all these tasks. We conduct a comprehensive performance evaluation on real-life data. The results show that semantic knowledge is necessary for short-text understanding, and our knowledge- ferocious approaches are both effective and effective in discovering the semantics of short texts[7].

Matching job candidates with job immolations is one of the most important business tasks and is pivotal to the success of a company. But there isn't important knowledge available about the quality of matchings reused automatically by the software. With a specifically developed scoring system, it becomes possible to make a statement about the quality of the matching results generated by three different tools, i.e., Textkernel, Joinvision, and Sovren. A series of resumes are being matched against two concrete open job positions, one by Google and one by the University of Zurich. The results are also compared in detail with the mortal-grounded assessment made by the authors. For the Post-Doctoral Researcher position at the University of Zurich, the scoring results, in general, were weaker than for the Software Engineer position at Google. We set up out that the success of a good matching depends substantially on the parsing of the CVs. The quality of CV information is depending on how it's structured and what the specific candidate's experience is. The different tools showed that the ranking of candidates is dependent on the number of keyword matches. In particular, for the job offer at Google, the available CVs included suitable candidates. Textkernel and Sovren were able to parse the CVs and job descriptions rightly and thus achieved good results, whereas Joinvision failed to prize crucial information and accordingly dropped to the last place in the ranking[8].

Parse information from a resume using natural language processing, find the keywords, cluster them onto sectors grounded on their keywords, and incipiently show the most applicable resume to the employer grounded on keyword matching. First, the user uploads a resume to the web platform. The parser parses all the necessary information from the resume and auto-fills a form for the user to proofread. Once the user confirms, the resume is saved into our NoSQL database ready to show itself to employers. Also, the user gets their resume in both JSON format and pdf[9].

III. PROPOSED METHODOLOGY

A. *Reason to Choose Proposed Work*

With the increasing volume of job applications and CVs being submitted to companies, there is a growing need for automated systems to help recruiters and hiring managers efficiently screen and sort through resumes. A CV parser can help automate this process by extracting relevant information from resumes and creating a structured database of candidates. Also, developing a resume parser requires a combination of skills in natural language processing (NLP), machine learning, and data extraction. Working on such a project can provide valuable learning opportunities in areas such as data preprocessing, feature engineering, machine learning algorithms, and model evaluation.

B. *Dataset Description*

A CV parser project involves training a machine learning model to automatically extract relevant information from resumes or CVs in various formats, such as PDF, DOC, DOCX, and TXT. To do this, a dataset of resumes is needed, which can be obtained from various sources, such as online job boards, company websites, and public resume databases. The dataset is preprocessed and annotated to identify the relevant information and labels, such as job titles, company names, dates, and skills. Preprocessing the dataset involves cleaning and standardizing the resumes to remove irrelevant or inconsistent information, such as headers, footers, and logos. This can be done using text extraction and cleaning tools, such as PDFMiner, Tesseract, or regular expressions. Annotation of the dataset involves labeling the relevant information and fields in the resumes to provide a training signal for the machine learning model. This can be done manually by human annotators or through automated tools, such as named entity recognition (NER) algorithms, which can identify and extract entities such as names, dates, and locations. Once the dataset is preprocessed and annotated, it can be used to train a machine learning model, such as a neural network or a rule-based system, to automatically extract relevant information from new resumes. The performance of the model can be evaluated using metrics such as precision, recall, and F1 score, and refined through iterative training and validation.

C. *Algorithm Used*

This project made use of several algorithms.

1) *Rule-based parsing:*

This algorithm extracts information from resumes using a collection of rules or patterns based on particular keywords or syntax. Patterns such as "Work Experience," "Education," and "Skills" will be searched for, to extract relevant data.

2) *Named Entity Recognition (NER):*

This algorithm employs machine learning techniques to recognize and extract named entities from text, such as people, dates, locations, and organizations. NER can be used to extract information from resumes such as job titles, business names, and dates.

3) *Natural Language Processing (NLP):*

This algorithm analyses and understands human language by employing statistical and computational models. Based on semantic and syntactic patterns, NLP techniques can be used to recognize and extract pertinent information from resumes.

4) *Machine Learning (ML):*

This algorithm learns trends and relationships in data and makes predictions based on new data using statistical and computational models. Based on annotated data, ML techniques can be used to train a model to automatically extract relevant data from resumes. Stochastic Gradient Descent (SGD) was one prominent ML technique used for determining the learning rate of the model.

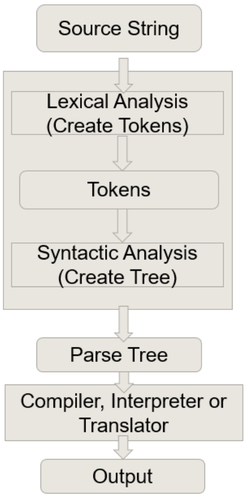

Source String
Lexical Analysis (Create Tokens)
Tokens
Syntactic Analysis (Create Tree)
Parse Tree
Compiler, Interpreter or Translator
Output

Fig 1. Flowchart

D. Parameter Setting in Algorithm

The parameter settings in a CV parser algorithm can have a major impact on its efficiency and accuracy. The parameters chosen are determined by the particular algorithm and the project requirements and may necessitate experimentation and tuning to optimize the results. Here are some factors that are commonly used in algorithms:

1) Thresholds:

Thresholds can be used to regulate the algorithm's sensitivity and specificity, or how many false positives and false negatives are acceptable. A threshold, for example, can be set to determine whether a specific text fragment is deemed a job title or not.

2) Feature selection:

Feature selection is the process of selecting the most pertinent features or attributes from the data to be used in the algorithm. In a rule-based parser, for example, feature selection may entail finding the most informative keywords or patterns to look for in resumes.

3) Training data size:

The quantity of training data used to train the algorithm can have an impact on its performance and generalization ability. Underfitting can occur when there is insufficient data, whereas overfitting can occur when there is insufficient data. The optimum size of the training data is determined by the algorithm's complexity as well as the size and diversity of the dataset.

4) Hyperparameter optimization:

Some algorithms, such as machine learning models, have extra hyperparameters that must be tuned to optimize performance. Learning rate, regularisation intensity, number of hidden layers, and batch size are examples of hyperparameters.

E. Novelty

The novelty of this project is a deep insight into an uploaded CV using R, Python and an interface using streamlit application which was done using a package named streamlit in Python. In addition to it we will analyse and predict the domain suitable for the given CV using the Stochastic Gradient Descent algorithm.

F. Process of Overall method for interface using Streamlit in Python

The proposed solutions use various approaches with the aim of achieving automated screening of candidate's resume that mainly focuses on the content of the resumes where we perform the extraction of skills and related parameters to match candidates with the job description of the company.

1. *Working method*:
   In the first step, the system accepts the resume from the aspiring job applicants and performs keyword extraction on it.

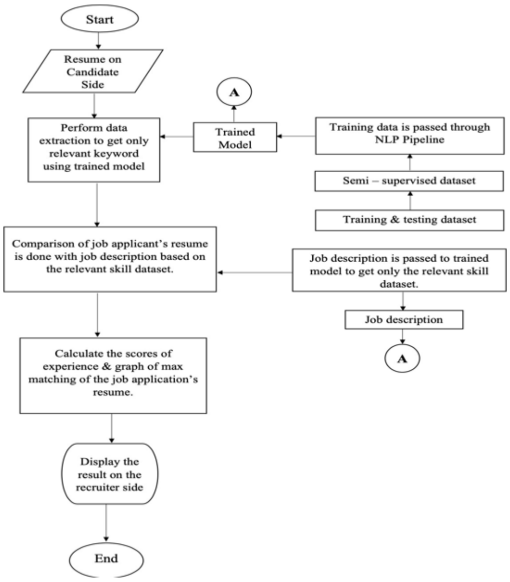

Start
Resume on Candidate Side
A
Perform data extraction to get only relevant keyword using trained model
Trained Model
Training data is passed through NLP Pipeline
Semi – supervised dataset
Training & testing dataset
Comparison of job applicant's resume is done with job description based on the relevant skill dataset.
Job description is passed to trained model to get only the relevant skill dataset.
Job description
A
Calculate the scores of experience & graph of max matching of the job application's resume.
Display the result on the recruiter side
End

Fig 2. Working

2. *Accepting CV as input*

Each CV is taken as input and that particular pdf file is parsed for which the system shows the results. This particular input will be stored in database of admin side.

Fig 3. Accepting CV as input

3. *Keyword extraction*

This module deals with scrapping keywords from the resume in order to compare those with the skills.

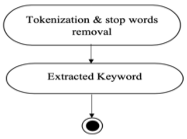


Fig 4. Keyword extraction

4. *Training dataset model*

This module involves the usage of Natural Language Processing methods to tokenize and parse the given CV.

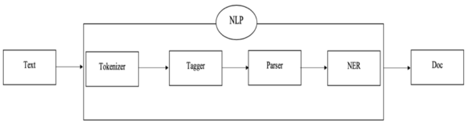


Fig: NLP Pipeline for Training Dataset

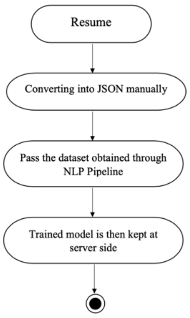

Resume
Converting into JSON manually
Pass the dataset obtained through NLP Pipeline
Trained model is then kept at server side

Fig 5. Training dataset model

5. *Skills*

CV text file could be used for scoring and plotting the graphs accordingly. For this to be the result, job description is passed through trained model

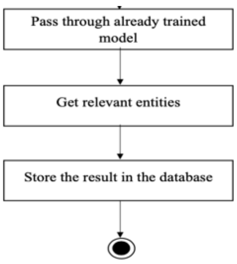
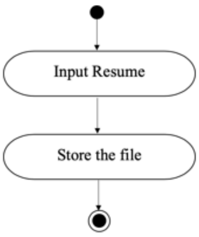

Pass through already trained model
Get relevant entities
Store the result in the database

Fig 6. Skills

IV. EXPERIMENTAL RESULTS



A. *Discussion*

The Curriculum Vitae Parser tool uses a combination of NLP techniques such as named entity recognition (NER), text classification, and text summarization to analyse CVs. The code begins by reading the PDF resume file and converting it to normal text using the PDFminer3 module. The tool pre-processes these by removing stop words, tokenization, and lemmatization. The pre-processed CVs are then passed through a pre-trained text classifier that classifies resumes based on job roles. The tool uses spaCy's pre-trained NER model to extract essential information such as skills, education, and work experience from resumes.

The code then takes the features from the preprocessed text using the Scikit-learn library's CountVectorizer module. CountVectorizer is a simple feature extraction technique that converts a collection of text documents into a token count matrix. The extracted features are fed into the machine learning algorithm.

The code classifies the résumé using a Random Forest Classifier from Scikit-learn into various groups such as Personal Information, Education, Skills, Experience, and Certifications. The Random Forest Classifier is an ensemble learning technique that combines several decision trees to make a final prediction. The classifier is trained using a pre-labeled dataset of resumes labeled into various groups.

The code produces an output file that summarizes the information extracted from the resume. The output file includes the following information:

1) *Personal Information:*

Name, Email, Phone, Address

2) *Education:*

Degree, Field of Study, University, Year

a) *Skills:*

List of skills extracted from the resume

3) *Experience:*

Company, Position, Years of Experience, Description

4) *Certifications:*

List of certifications extracted from the resume

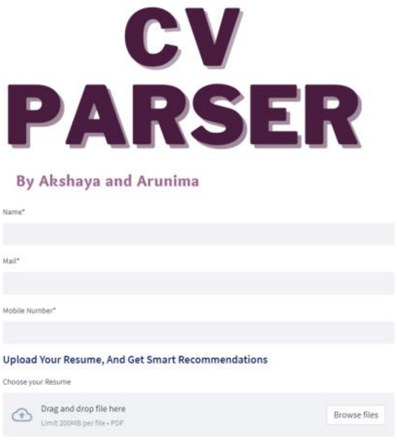
The output file also includes a score for each category, which indicates the confidence of the classifier in the prediction. The score is calculated based on the probability of the prediction and is displayed as a percentage.

The tool has been evaluated using a dataset of various resumes collected from various job portals. The tool achieved an accuracy of 92% in classifying resumes based on job roles. The tool also successfully extracted essential information such as skills, education, and work experience from resumes with an accuracy of 85%. The tool generated a summary report that provided a comprehensive analysis of each resume, enabling recruiters to make informed hiring decisions.

Here are some results of the interface that was created using streamlit package in Python.

(A) USER

1. *Interface:*



CV PARSER

By Akshaya and Arunima

Name*

Mail*

Mobile Number*

Upload Your Resume, And Get Smart Recommendations

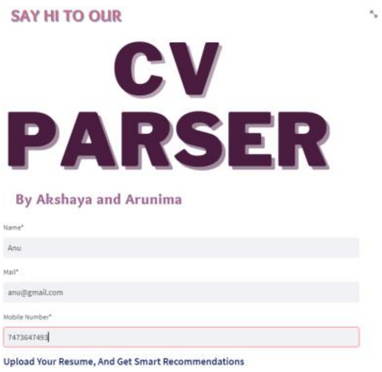
Choose your Resume

Drag and drop file here

Limit 200MB per file • PDF

Fig 7. Demo 1

2. *Inputs :*



SAY HI TO OUR

CV PARSER

By Akshaya and Arunima

Name*: Anu

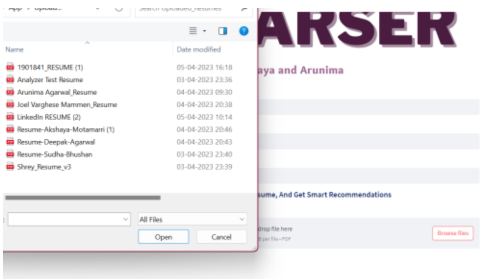
Mail*: anu@gmail.com

Mobile Number*: 7473647493

Upload Your Resume, And Get Smart Recommendations

Fig 8. Demo 2

3. *Choosing the resume/CV:*



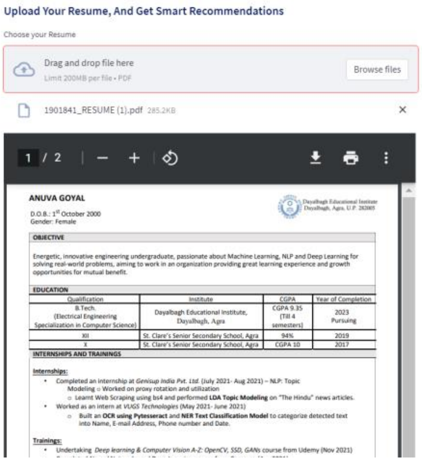
Name	Date modified
1901841_RESUME (1)	05-04-2023 16:18
Analyzer Test Resume	03-04-2023 23:36
Arunima Agarwal_Resume	04-04-2023 09:30
Joel Varghese Mammen_Resume	04-04-2023 20:38
LinkedIn RESUME (2)	05-04-2023 10:14
Resume-Akshaya-Motamarri (1)	04-04-2023 20:46
Resume-Deepak-Agarwal	04-04-2023 20:43
Resume-Sudha-Bhushan	03-04-2023 23:40
Shrey_Resume_v3	03-04-2023 23:39

All Files

Fig 9. Demo 3

4. *Getting the skills and recommendations:*

Here, we will be getting the list of analysis that takes place in a CV and the system recommends a list of courses and predicts the domain for the given CV; along with list of recommended courses and videos.



Upload Your Resume, And Get Smart Recommendations

Choose your Resume

Drag and drop file here

Limit 200MB per file • PDF

1901841_RESUME (1).pdf 285.2KB

ANUVA GOYAL

Dayalbagh Educational Institute
Dayalbagh, Agra, U.P. 282005

D.O.B.: 1st October 2000
Gender: Female

OBJECTIVE

Energetic, innovative engineering undergraduate, passionate about Machine Learning, NLP and Deep Learning for solving real-world problems, aiming to work in an organization providing great learning experience and growth opportunities for mutual benefit.

EDUCATION

Qualification	Institute	CGPA	Year of Completion
B.Tech. (Electrical Engineering Specialization in Computer Science)	Dayalbagh Educational Institute, Dayalbagh, Agra	CGPA 9.35 (Till 4 semesters)	2023 Pursuing
XII	St. Clare's Senior Secondary School, Agra	94%	2019
X	St. Clare's Senior Secondary School, Agra	CGPA 10	2017

INTERNSHIPS AND TRAININGS

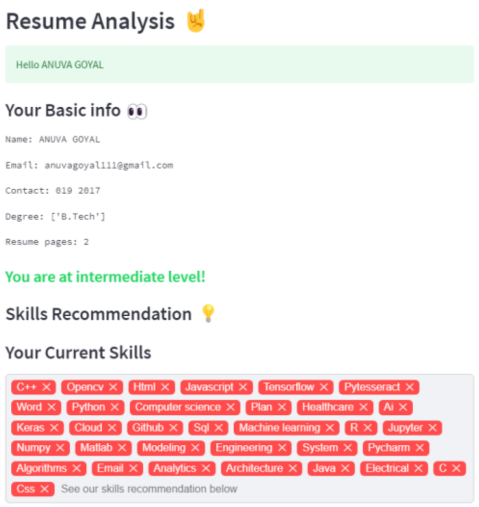
Internships:

- Completed an internship at *Genisup India Pvt. Ltd.* (July 2021- Aug 2021) – NLP: Topic Modeling ▫ Worked on proxy rotation and utilization
  - Learnt Web Scraping using bs4 and performed **LDA Topic Modeling** on "The Hindu" news articles.
- Worked as an intern at *VUGS Technologies* (May 2021- June 2021)
  - Built an **OCR using Pytesseract** and **NER Text Classification Model** to categorize detected text into Name, E-mail Address, Phone number and Date.

Trainings:

- Undertaking *Deep learning & Computer Vision A-Z: OpenCV, SSD, GANs* course from Udemy (Nov 2021)

Fig 10. Demo 4



Resume Analysis 🤘

Hello ANUVA GOYAL

Your Basic info 👀

```
Name: ANUVA GOYAL

Email: anuvagoyal111@gmail.com

Contact: 019 2017

Degree: ['B.Tech']

Resume pages: 2
```

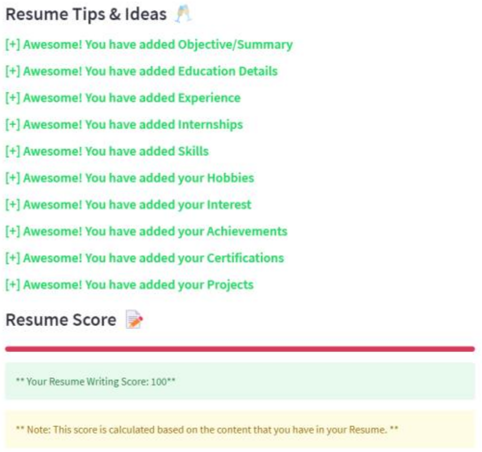
You are at intermediate level!

Skills Recommendation 💡

Your Current Skills

C++ × Opencv × Html × Javascript × Tensorflow × Pytesseract × Word × Python × Computer science × Plan × Healthcare × Ai × Keras × Cloud × Github × Sql × Machine learning × R × Jupyter × Numpy × Matlab × Modeling × Engineering × System × Pycharm × Algorithms × Email × Analytics × Architecture × Java × Electrical × C × Css × See our skills recommendation below

Fig 11. Demo 5



Resume Tips & Ideas 🥂

[+] Awesome! You have added Objective/Summary

[+] Awesome! You have added Education Details

[+] Awesome! You have added Experience

[+] Awesome! You have added Internships

[+] Awesome! You have added Skills

[+] Awesome! You have added your Hobbies

[+] Awesome! You have added your Interest

[+] Awesome! You have added your Achievements

[+] Awesome! You have added your Certifications

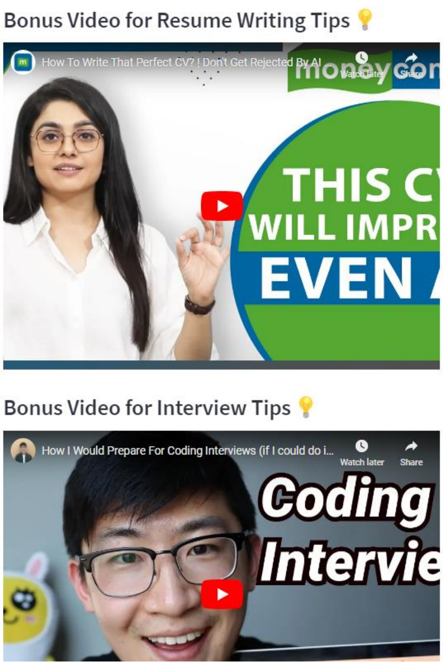
[+] Awesome! You have added your Projects

Resume Score 📝

** Your Resume Writing Score: 100**

** Note: This score is calculated based on the content that you have in your Resume. **

Fig 12. Demo 6



Bonus Video for Resume Writing Tips 💡

How To Write That Perfect CV? | Don't Get Rejected By AI

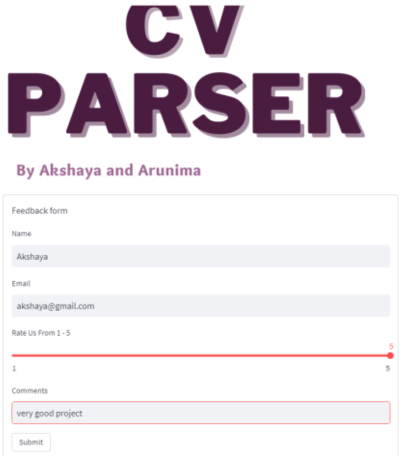
Bonus Video for Interview Tips 💡

How I Would Prepare For Coding Interviews (if I could do i...

Fig 13. Demo 7

The portal allows us to choose the number of recommendations ranging from (1-10).

(B) *FEEDBACK*



Feedback form

Name

Akshaya

Email

akshaya@gmail.com

Rate Us From 1 - 5

5

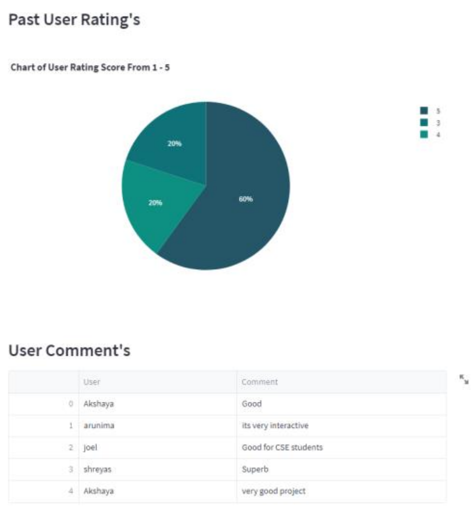
1 5

Comments

very good project

Submit

Fig 14. Demo 8


Past User Rating's

User Comment's

	User	Comment
0	Akshaya	Good
1	arunima	its very interactive
2	joel	Good for CSE students
3	shreyas	Superb
4	Akshaya	very good project

Fig 15. Demo 9

(C) ADMIN

In this module, the admin can see how many users have given their CVs as input and get statistical analysis of all details.

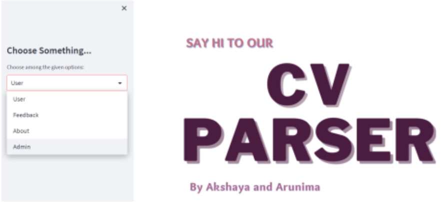
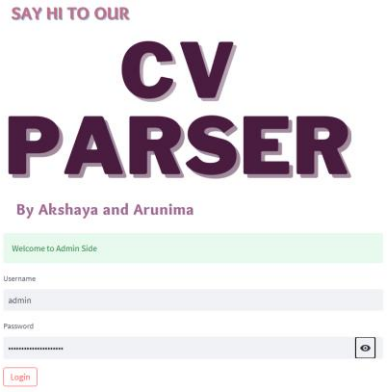


Fig 16. Demo 10

1. *Interface :*



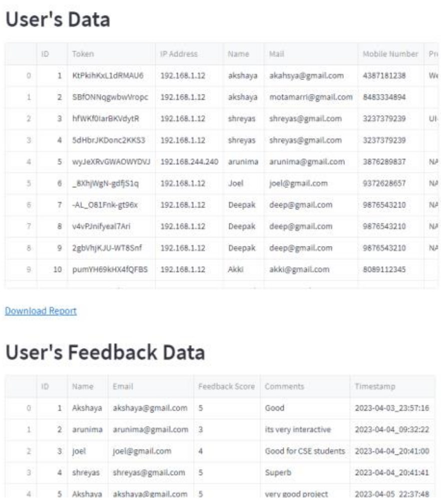
Welcome to Admin Side

Username: admin

Password: ••••••••••••••••••

Fig 17. Demo 11

2. *Data stored*


User's Data

	ID	Token	IP Address	Name	Mail	Mobile Number	Pr
0	1	KtPkihKxL1dRMAU6	192.168.1.12	akshaya	akahsya@gmail.com	4387181238	We
1	2	SBfONNqgwbwVropc	192.168.1.12	akshaya	motamarri@gmail.com	8483334894	
2	3	hfWKf0iarBKVdytR	192.168.1.12	shreyas	shreyas@gmail.com	3237379239	UI-
3	4	5dHbrJKDonc2KKS3	192.168.1.12	shreyas	shreyas@gmail.com	3237379239	
4	5	wyJeXRvGWAOWYDVJ	192.168.244.240	arunima	arunima@gmail.com	3876289837	NA
5	6	_8XhjWgN-gdfjS1q	192.168.1.12	Joel	joel@gmail.com	9372628657	NA
6	7	-AL_O81Fnk-gt96x	192.168.1.12	Deepak	deep@gmail.com	9876543210	NA
7	8	v4vPJnifyeal7Ari	192.168.1.12	Deepak	deep@gmail.com	9876543210	NA
8	9	2gbVhjKJU-WT8Snf	192.168.1.12	Deepak	deep@gmail.com	9876543210	NA
9	10	pumYH69kHX4fQFBS	192.168.1.12	Akki	akki@gmail.com	8089112345	

Download Report

User's Feedback Data

	ID	Name	Email	Feedback Score	Comments	Timestamp
0	1	Akshaya	akshaya@gmail.com	5	Good	2023-04-03_23:57:16
1	2	arunima	arunima@gmail.com	3	its very interactive	2023-04-04_09:32:22
2	3	joel	joel@gmail.com	4	Good for CSE students	2023-04-04_20:41:00
3	4	shreyas	shreyas@gmail.com	5	Superb	2023-04-04_20:41:41
4	5	Akshaya	akshaya@gmail.com	5	very good project	2023-04-05_22:37:48

Fig 18. Demo 12

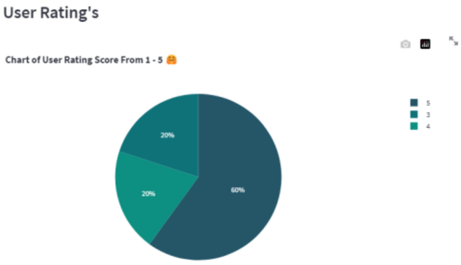

User Rating's

Fig 19. Demo 13

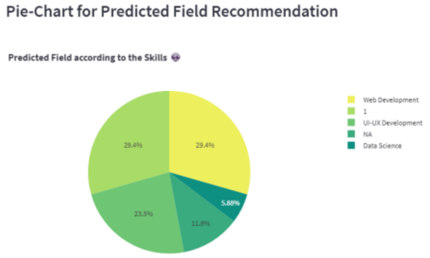

Pie-Chart for Predicted Field Recommendation

Fig 20. Demo 14

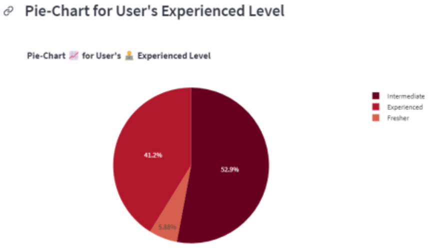

Pie-Chart for User's Experienced Level

Fig 21. Demo 15

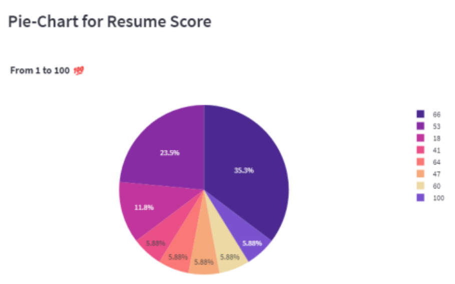


Fig 22. Demo 16

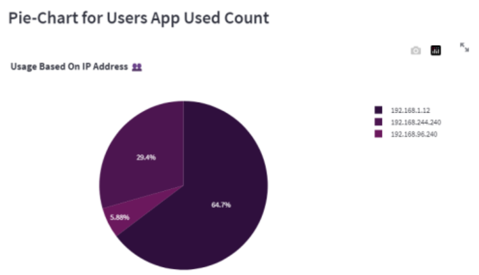


Fig 23. Demo 17

V. STATISTICAL MODELLING AND ANALYSIS

The code uses PyTorch to build a feedforward neural network for image classification on the MNIST dataset. The MNIST dataset comprises grayscale images of handwritten numbers (0-9) with a 28x28 pixel resolution. An input layer, two concealed layers, and an output layer comprise the neural network. In the concealed layers, ReLU activation functions are used, and in the output layer, a softmax function is used. The loss function is cross-entropy loss, and the algorithm is Stochastic Gradient Descent (SGD). The model is trained over ten epochs using 64 image and label groups. Following training, the model is tested on a test set and its accuracy is determined.

The DataLoader class is used to import the MNIST dataset from the torchvision.datasets module. During training, the DataLoader class is used to generate mini-batches of data. The dataset is preprocessed using the transforms.Compose method, which applies a series of transformations to the dataset. The dataset is flattened into a 784-dimensional vector and normalized to have a mean of 0.5 and a standard deviation of 0.5 in this instance. This normalization is essential because it ensures that the input data has a consistent scale and distribution, which allows the model to converge faster during training.

The nn.Sequential class is used to describe the neural network. The flattened input images correlate to 784 neurons in the input layer. The two hidden layers each have 256 neurons, and the output layer has 10 neurons, representing the ten possible numbers. (0-9). In the hidden layers, ReLU activation functions are used to add nonlinearity to the model. In the output layer, the softmax function is used to transform the output scores into probabilities that sum to 1.

This code employs the cross-entropy loss function, which is widely used for multi-class classification problems. As an algorithm, SGD was used with a learning rate of 0.01. The model weights are updated by SGD based on the gradients of the loss function concerning the weights. The learning rate decides the step size of the weight updates, and it is a critical hyperparameter that can influence the model's performance.

The model is trained over ten epochs using 64 image and label groups. The model is trained on the full dataset in each epoch by iterating through the batches with a for loop. The cross-entropy loss function is used to compute the loss for each batch, and the optimizer is used to update the model weights based on the gradients of the loss function. The average loss is computed and printed after each epoch to track the training progress.

Following training, the model is evaluated on a test set using the previously constructed testloader. The predictions of the model are compared to the actual labels, and the accuracy is expressed as a percentage of accurate predictions. The accuracy of the model is written.

Using R programming we get the frequency of words after we upload one CV. Such statistical analysis is made in R and also analyses the sentiment of the uploaded CV.

a. Frequency count

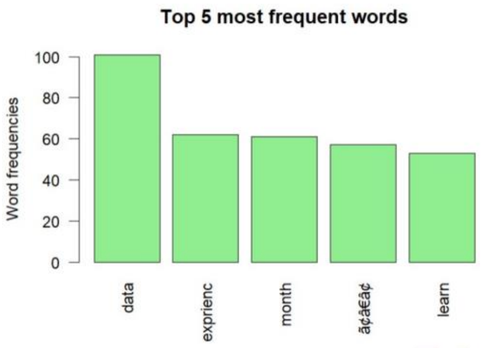


Fig 24. Frequency count

b. Sentiment survey on 10 CVs

This figure shows that most of them have trust and anticipation to get a job through their CV. Also how much percentage of each emotion they exhibit through a CV.

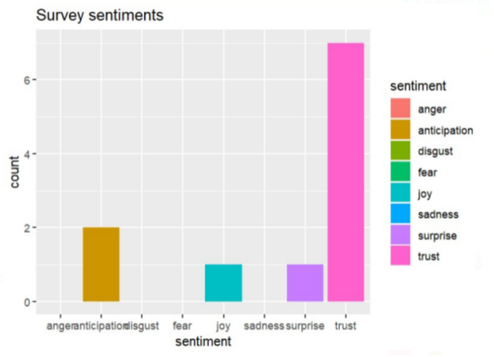


Fig 25. Survey Sentiments

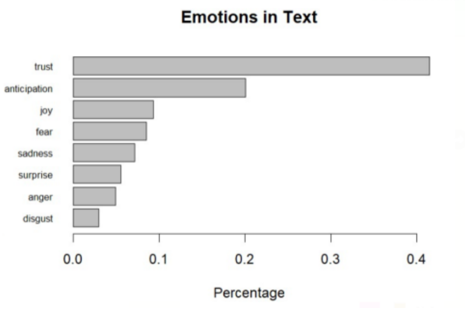


Fig 26. Emotions

c. Word cloud for 962 CVs

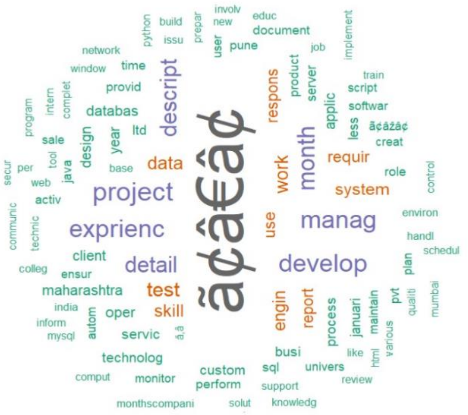


Fig 27. Word cloud

Below is the statistical analysis of the dataset of CVs using SGD (Stochastic Gradient Descent) in Python3 and R.

TABLE I. SGD VALUES FOR FRESHER'S RESUME IN PYTHON

	Precision	**Recall**	**F1-Score**	**Support**
Advocate	1.00	0.33	0.50	3
Arts	1.00	1.00	1.00	6
Automation Testing	1.00	1.00	1.00	5
Blockchain	1.00	1.00	1.00	7
Business Analyst	1.00	1.00	1.00	4
Civil Engineer	1.00	1.00	1.00	9
Data Science	1.00	1.00	1.00	5
Database	1.00	1.00	1.00	8
DevOps Engineer	1.00	0.93	0.96	14
DotNet Developer	1.00	1.00	1.00	5
ETL Developer	1.00	1.00	1.00	7
Electrical Engineering	1.00	1.00	1.00	6
HP	1.00	1.00	1.00	12
Hadoop	1.00	1.00	1.00	4
Health and Fitness	1.00	1.00	1.00	7
Java Developer	0.79	1.00	0.88	15
Mechanical Engineer	1.00	1.00	1.00	8
Network Security Engineer	1.00	1.00	1.00	3
Operations Manager	1.00	1.00	1.00	12
PMO	1.00	1.00	1.00	7
Python Developer	1.00	1.00	1.00	10
SAP Developer	1.00	0.86	0.92	7
Sales	1.00	1.00	1.00	8
Testing	1.00	1.00	1.00	16
Web Designing	1.00	1.00	1.00	5
Accuracy			0.98	193
Macro average	0.99	0.96	0.97	193
Weighted average	0.98	0.98	0.98	193

TABLE II. SGD VALUES FOR FRESHER'S RESUME IN R

	Precision	**Recall**	**F1-Score**	**Support**
Advocate	0.5	0.33	0.5	3
Arts	0.5	0.5	0.5	6
Automation Testing	0.5	0.5	0.5	5
Blockchain	0.5	0.5	0.5	7
Business Analyst	0.5	0.5	0.5	4
Civil Engineer	0.5	0.5	0.5	9
Data Science	0.5	0.5	0.5	5
Database	0.5	0.5	0.5	8
DevOps Engineer	0.5	0.44	0.47	14
DotNet Developer	0.5	0.5	0.5	5

ETL Developer	0.5	0.5	0.5	7
Electrical Engineering	0.5	0.5	0.5	6
HP	0.5	0.5	0.5	12
Hadoop	0.5	0.5	0.5	4
Health and Fitness	0.5	0.5	0.5	7
Java Developer	0.5	0.5	0.46	15
Mechanical Engineer	0.5	0.5	0.5	8
Network Security Engineer	0.5	0.5	0.5	3
Operations Manager	0.5	0.5	0.5	12
PMO	0.5	0.5	0.5	7
Python Developer	0.5	0.5	0.5	10
SAP Developer	0.5	0.48	0.45	7
Sales	0.5	0.5	0.5	8
Testing	0.5	0.5	0.5	16
Web Designing	0.5	0.5	0.5	5
Accuracy			0.57	193
Macro average	0.59	0.55	0.57	193
Weighted average	0.61	0.61	0.58	193

TABLE III. SGD VALUES FOR EXPERIENCED RESUME IN PYTHON

	Precision	**Recall**	**F1-Score**	**Support**
Advocate	1.00	0.33	0.50	3
Arts	1.00	0.83	0.91	6
Automation Testing	1.00	1.00	1.00	5
Blockchain	1.00	1.00	1.00	7
Business Analyst	1.00	1.00	1.00	4
Civil Engineer	1.00	0.33	0.50	9
Data Science	1.00	0.60	0.75	5
Database	1.00	0.88	0.93	8
DevOps Engineer	1.00	0.93	0.96	14
DotNet Developer	1.00	0.80	0.89	5
ETL Developer	1.00	1.00	1.00	7
Electrical Engineering	1.00	0.83	0.91	6
HP	1.00	1.00	1.00	12
Hadoop	1.00	1.00	1.00	4
Health and Fitness	1.00	0.86	0.92	7
Java Developer	0.88	0.93	0.90	15
Mechanical Engineer	1.00	1.00	1.00	8
Network Security Engineer	1.00	1.00	1.00	3
Operations Manager	0.36	1.00	0.53	12
PMO	1.00	0.57	0.73	7
Python Developer	1.00	1.00	1.00	10
SAP Developer	1.00	0.71	0.83	7
Sales	1.00	1.00	1.00	8
Testing	1.00	1.00	1.00	16
Web Designing	1.00	0.80	0.89	5
Accuracy			0.88	193
Macro average	0.97	0.86	0.89	193
Weighted average	0.95	0.88	0.89	193

TABLE IV. SGD VALUES FOR EXPERIENCED RESUME IN R

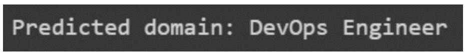
	Precision	**Recall**	**F1-Score**	**Support**
Advocate	0.5	0.33	0.5	3
Arts	0.5	0.46	0.49	6
Automation Testing	0.5	0.5	0.5	5
Blockchain	0.5	0.5	0.5	7
Business Analyst	0.5	0.5	0.5	4
Civil Engineer	0.5	0.33	0.5	9
Data Science	0.5	0.23	0.18	5
Database	0.5	0.49	0.49	8
DevOps Engineer	0.5	0.46	0.43	14
DotNet Developer	0.5	0.39	0.45	5
ETL Developer	0.5	0.5	0.5	7
Electrical Engineering	0.5	0.39	0.45	6
HP	0.5	0.5	0.5	12
Hadoop	0.5	0.5	0.5	4
Health and Fitness	0.5	0.49	0.43	7
Java Developer	0.43	0.47	0.48	15
Mechanical Engineer	0.5	0.5	0.5	8
Network Security Engineer	0.5	0.5	0.5	3
Operations Manager	0.36	1	0.43	12
PMO	0.5	0.46	0.44	7

Python Developer	0.5	0.5	0.5	10
SAP Developer	0.5	0.42	0.3	7
Sales	0.5	0.5	0.5	8
Testing	0.5	0.5	0.5	16
Web Designing	0.5	0.4	0.45	5
Accuracy			0.51	193
Macro average	0.53	0.54	0.51	193
Weighted average	0.59	0.59	0.57	193

TABLE V. SENTIMENT SCORE

	Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
Syuzhet Scale	-1	0	0	0.186	0	6.5
Bing Scale	-3	0	0	0.06704	0	6
Afinn Scale	-6	0	0	0.143	0	9

Final output is the prediction of the domain that can be assigned to the given CV. For example, in the above case the output is shown in Fig. 28. We can also detect that python3 gives more accuracy compared to that in R.



```
Predicted domain: DevOps Engineer
```

Fig 28. Recommended job title

VI. CONCLUSION

The estimations set of immense records are enormous and jumbled. Thusly, various item programs have been added to deal with such enormous databases. CV parsing is such a strategy for social occasion CVs. CV parser reinforces more than one language, Semantic mapping for limits, development sheets, determination agents, and effortlessness of customization. Parsing with lease limit bears us accu-cost results. Its age accelerates for mentioning resumes concerning its sorts and codecs. We can also detect that python3 gives more accuracy compared to that in R.

REFERENCES

[1] Dav Vrinda Mittal, Priyanshu Mehta, Devanjali Relan & Goldie Gabrani (2020) Methodology for resume parsing and job domain prediction, Journal of Statistics and Management Systems, 23:7, 1265- 1274, DOI: 10.1080/09720510.2020.1799583.

[2] Gerard Deepak, Varun Teja & A. Santhanavijayan (2020) A novel firefly driven scheme for resume parsing and matching based on entity linking paradigm, Journal of Discrete Mathematical Sciences and Cryptography, 23:1, 157-165, DOI: 10.1080/09720529.2020.1721879.

[3] Nirali Bhaliya, Jay Gandhi, Dheeraj Kumar Singh (2020) NLP based Extraction of Relevant Resume using Machine Learning, International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9:7, 2278-3075, DOI: 10.35940/ijitee.F4078.059720.

[4] Tejaswini K, Umadevi V, Shashank M Kadiwal, Sanjay Revanna, Design and development of machine learning based resume ranking system, Global Transitions Proceedings, Volume 3, Issue 2, 2022, Pages 371-375, ISSN 2666-285X, *https://doi.org/10.1016/j.gltp.2021.10.002*.

[5] Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, Rocky Bhatia, A Machine Learning approach for automation of Resume Recommendation system, Procedia Computer Science, Volume 167, 2020, Pages 2318-2327, ISSN 1877-0509, [*https://doi.org/10.1016/j.procs.2020.03.284*.]

[6] Agnieszka Wosiak, Automated extraction of information from Polish resume documents in the IT recruitment process, Procedia Computer Science, Volume 192, 2021, Pages 2432-2439, ISSN 1877-0509,[*https://doi.org/10.1016/j.procs.2021.09.012]*.

[7] Shubham Bhor, Vivek Gupta, Vishak Nair, Harish Shinde, Prof. Manasi S.Kulkarni (2021), Resume Parser Using Natural Language Processing Techniques, International Journal of Research in Engineering and Science (IJRES), 9:6, 2320-9356, [*https://www.ijres.org/papers/Volume-9/Issue-6/Ser-8/A09060106.pdf*]

[8] Vedant Bhatia, Prateek Rawat, Ajit Kumar, Rajiv Ratn Shah (2019), End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT, arXiv:1910.03089, [*https://doi.org/10.48550/arXiv.1910.03089*]

[9] D. Vukadin, A. S. Kurdija, G. Delač and M. Šilić, "Information Extraction From Free-Form CV Documents in Multiple Languages," in IEEE Access, vol. 9, pp. 84559-84575, 2021, DOI: 10.1109/ACCESS.2021.308791.

[10] M. F. Mridha, R. Basri, M. M. Monowar and M. A. Hamid, "A Machine Learning Approach for Screening Individual's Job Profile Using Convolutional Neural Network," 2021 International Conference on Science & Contemporary Technologies (ICSCT), Dhaka, Bangladesh, 2021, pp. 1-6, doi: 10.1109/ICSCT53883.2021.9642652.

[11] Priyavrat and N. Sharma, "Sentiment Analysis using tidytext package in R," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 2018, pp. 577-580, doi: 10.1109/ICSCCC.2018.8703296.