Abstract

Talent acquisition is most import task for the success of the company. In current situation for a given job thousands of job seeker apply which make hard for the hiring team to go through each and every resume manually and check for the credibility of the applicant. Similar for the job seeker in the large market thousands of the jobs are available which makes finding suitable job difficult for the user. This project aims to solve this problem by making automation of the resume matching process by using various technique for the data extraction from the given text or description and finding similarity between the job seeker's profile and job description. Similarity can be found using the cosine similarity or Euclidean distance. Also data can be extracted from the resume to finding keywords in bags of words.

Keywords: Recommendation System - Job Recommendation System - Hybrid approach - Content based Filtering - Cosine similarity

Contents

List of Figures

List Of Acronyms

CF Collaborative filtering

CBF Content-based filtering

RS Recommendation System

JRS Job Recommender Systems

SVM Support Vector Machine

List of Symbols

$\sqrt{}$ Square root

θ Angle between vector

$\sum$ Summation

$\cdot$ Dot product

Chapter 1

Introduction

More and more applications have been broadly developed, and new techniques have emerged to support human decisions suggesting services, products, and various types of information to customers. One field of research in this direction is that of Recommender Systems. Recommender Systems use various techniques and algorithms to isolate irrelevant information from a vast amount of data and generate personalized suggestions of a small subset of them that a user can examine in a reasonable amount of time. The increasing usage of the Internet has heightened the need for online job hunting. The critical problem is that most job-hunting websites display recruitment information to website viewers. Many websites on the Internet give employment opportunities, but the task is tedious as students need to go through a large amount of information, taking lots of time and energy and suffering from unwanted or less helpful information. Jobseekers have to retrieve all the information to find jobs they want to apply for. The whole procedure is tedious and inefficient. One field of research in this direction is that of Recommender Systems.

1.1 Applications

Dealing with the tremendous amount of recruiting information over the Internet, a job seeker always spends hours finding useful ones. With a huge number of different job roles existing today along with the typically large number of applications received, short-listing poses a challenge for the human resource department. This is only further worsened by the lack of diverse skill and domain knowledge within the HR department, required for effective screening. Being able to weed out non-relevant profiles as early as possible in the pipeline results in cost savings, both in terms of time as well as money.[4]

1.2 Motivation

Talent acquisition is an important, crucial, complex, essential task in industries that requires a significant amount of time. Talent acquisition has the most challenging part. The lack of a standard structure and format for a resume makes a short listing of desired profiles for required roles very tedious and time-consuming. Effective screening of resumes requires domain knowledge to understand the relevance and applicability of a profile for the job role. With a massive number of different job roles existing today and the typically large number of applications received, short-listing challenges the human resource department.

In addition, in most Recommendation System (RS)s, the most general application of recommendation algorithms uses Collaborative filtering (CF) algorithms without considering the user's resume and job description. That means candidates' resumes and details of recruiting information. So we proposed an improved algorithm based on Content-based filtering (CBF). Our aim is to give an effective method of recommendation for online job hunting and talent hunting. We hope to offer candidates a personalized service that can help them find ideal jobs quickly and conveniently.

1.3 Objectives

The e-recruiting platforms are usually based on Boolean search and filtering techniques that cannot sufficiently capture the complexity of a person-job fit as selection decisions. Much literature has applied the recommender system concept to the job problem. Recommendation between entities of the domain: users and opportunities

The job recommendation problem is a bidirectional recommendation between job-seeker and job.[2] Two viewpoints are distinguished: from recruiters and job seekers. The recruiters generate the job description by determining the set of requirements and constraints on skills, expertise levels, and degrees. The job-seeker, on the other hand, generates the candidate's resume by specifying the academic background, previous work experience and skills[2].

Based on the requirement that a good match between jobs and persons needs to take into account both the preferences of the candidate and the preferences of the recruiter to recommend the job.

1.4 Contribution

We are going to job seeker's skill and preferences for job and the job description in to the consider to recommend the job for that we need to find the similarity between the job seeker's profile and job's profile. So, we can create a vector of words containing the keywords for the job and later we can use any similarity algorithm like cosine similarity to recommend the job to the job seeker.

1.5 Organization of project report

This chapter covers the introduction to the project along with its application, motivation, objective and overview. Chapter 2 presents a theoretical background of the terminologies in the job recommendation as well as other important concepts needed to understand the project better. Chapter 3 is the Literature Survey which summarises the work done in the job recommendation to recommend the job. Later in Chapter 3, a review of various job recommendation algorithms is discussed. Our proposed methodology and logic development of the same is covered in Chapter 4. Chapter 5 discusses the brief overview about our data. Chapter 7 ends the report with conclusion and future work proposed.

Chapter 2

Theoretical Background

This chapter discuss various recommendation technique and how this technique works and the advantages and limitation of the various recommendation technique. Finally, it contains overview of different RS.

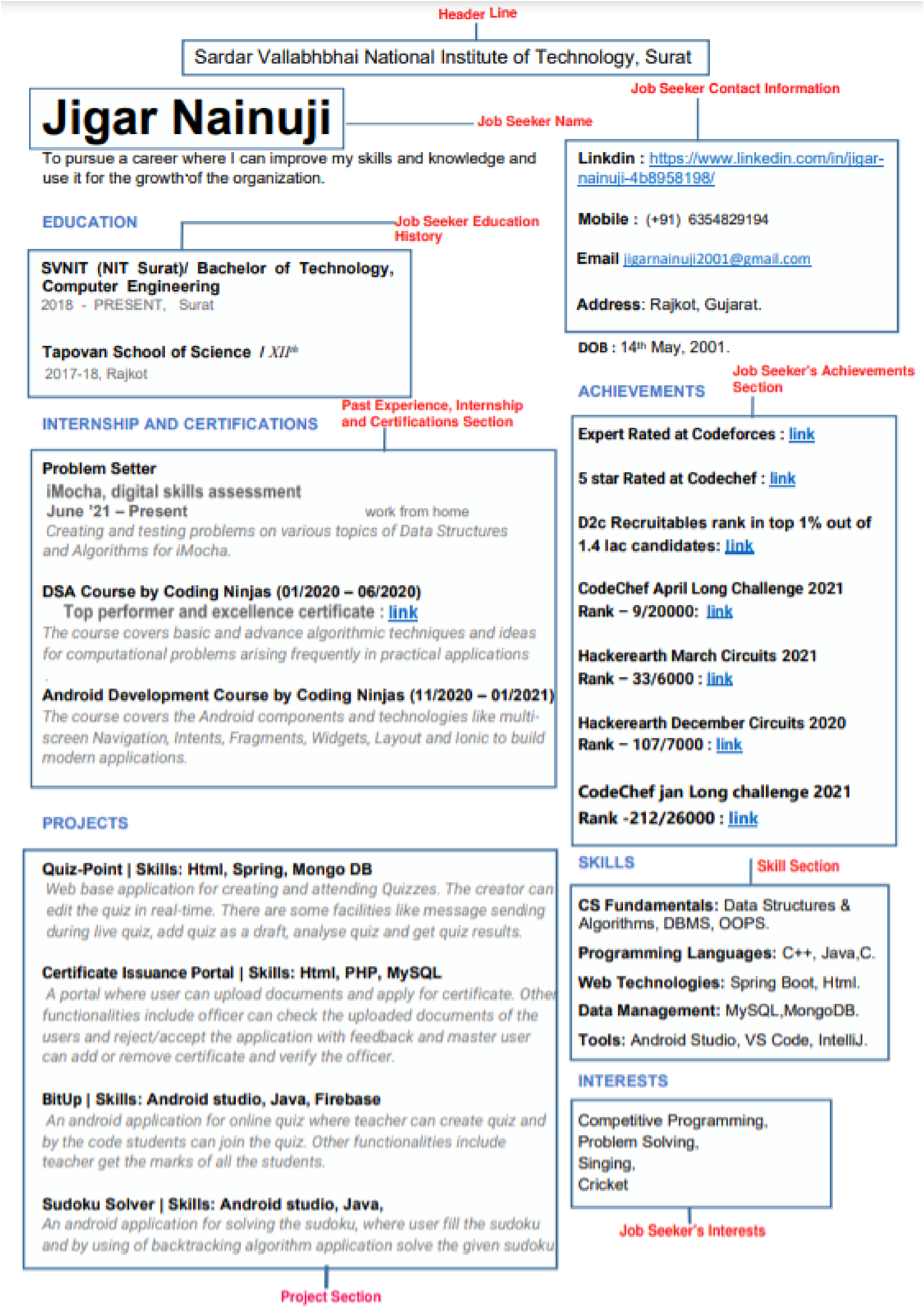
2.1 Resume

A resume is a formal document created by a job seeker to list their qualifications for a particular position. A customized cover letter is typically sent with a resume, in which the candidate expresses interest in a specific job or organization and highlights critical information on the CV.

Skills :
knowledge of different technologies in which job seeker have experience or he/she learn that
Language: java, python c++,c, HTML
Technologies: Spring boot, Django, node js.

Past Experience, Internships and Certification :
Job seeker's experience in the previous companies as a full-time worker or an intern. Job Seeker does certificates of Internships and different courses.

Includes the List of the companies that Job Seeker worked for, employment/internship dates, their positions, and brief descriptions of their work responsibilities, enriched with keywords and enhanced with bulleted lists of quantifiable achievements done in Job / Internship. It also includes a List Of certifications that a Jobseeker has.



Header Line
Job Seeker Contact Information
Job Seeker Name
Job Seeker Education History
Job Seeker's Achievements Section
Past Experience, Internship and Certifications Section
Skill Section
Job Seeker's Interests
Project Section

Sardar Vallabhbhai National Institute of Technology, Surat

Jigar Nainuji

To pursue a career where I can improve my skills and knowledge and use it for the growth of the organization.

EDUCATION

SVNIT (NIT Surat)/ Bachelor of Technology, Computer Engineering
2018 - PRESENT, Surat

Tapovan School of Science / *XII*th
2017-18, Rajkot

INTERNSHIP AND CERTIFICATIONS

Problem Setter
iMocha, digital skills assessment
June '21 – Present work from home
Creating and testing problems on various topics of Data Structures and Algorithms for iMocha.

DSA Course by Coding Ninjas (01/2020 – 06/2020)
Top performer and excellence certificate : link
The course covers basic and advance algorithmic techniques and ideas for computational problems arising frequently in practical applications

Android Development Course by Coding Ninjas (11/2020 – 01/2021)
The course covers the Android components and technologies like multi-screen Navigation, Intents, Fragments, Widgets, Layout and Ionic to build modern applications.

PROJECTS

Quiz-Point | Skills: Html, Spring, Mongo DB
Web base application for creating and attending Quizzes. The creator can edit the quiz in real-time. There are some facilities like message sending during live quiz, add quiz as a draft, analyse quiz and get quiz results.

Certificate Issuance Portal | Skills: Html, PHP, MySQL
A portal where user can upload documents and apply for certificate. Other functionalities include officer can check the uploaded documents of the users and reject/accept the application with feedback and master user can add or remove certificate and verify the officer.

BitUp | Skills: Android studio, Java, Firebase
An android application for online quiz where teacher can create quiz and by the code students can join the quiz. Other functionalities include teacher get the marks of all the students.

Sudoku Solver | Skills: Android studio, Java,
An android application for solving the sudoku, where user fill the sudoku and by using of backtracking algorithm application solve the given sudoku

Linkdin : https://www.linkedin.com/in/jigar-nainuji-4b8958198/

Mobile : (+91) 6354829194

Email jigarnainuji2001@gmail.com

Address: Rajkot, Gujarat.

DOB : 14th May, 2001.

ACHIEVEMENTS

Expert Rated at Codeforces : link

5 star Rated at Codechef : link

D2c Recruitables rank in top 1% out of 1.4 lac candidates: link

CodeChef April Long Challenge 2021
Rank – 9/20000: link

Hackerearth March Circuits 2021
Rank – 33/6000 : link

Hackerearth December Circuits 2020
Rank – 107/7000 : link

CodeChef jan Long challenge 2021
Rank -212/26000 : link

SKILLS

CS Fundamentals: Data Structures & Algorithms, DBMS, OOPS.

Programming Languages: C++, Java,C.

Web Technologies: Spring Boot, Html.

Data Management: MySQL,MongoDB.

Tools: Android Studio, VS Code, IntelliJ.

INTERESTS

Competitive Programming,
Problem Solving,
Singing,
Cricket

Figure 2.1: Resume Block

Projects :
Current students or recent graduates can use university projects to highlight their relevant skills in a more practical setting. For others, it can include freelance projects and personal projects also.

Achievements :
Includes List of Achievements that a job seeker has achieved. Achievements are things they did that had a lasting impact on a company or client. It is a result that they bring about while fulfilling a particular role.

Education :
Includes the job seeker's educational background. Employers look for a few essential aspects of resumes, including education. This information will give interviewers a better idea of their background, which might help them figure out if they are a good fit for the job.

Contact :
Include Job seeker's full name, street address, city, state, and zip code. Also, include their phone number and email address. If they have a LinkedIn profile or personal website, include these URLs in their contact section.

Interest :
Includes a job seeker's shared interests. The themes and broad ideas that you appreciate in your daily life are considered interests. They are usually more all-encompassing concepts that you are enthusiastic about. Interests are broad notions that guide your real-life decisions and activities.

Tools :
Includes Common tools That a job seeker used in the past or wants to work with them.

Personal profile :
Includes Job seeker personal information like Contact Number, Email id, URLs of portfolios, and Linkedin profiles.

2.2 Overview of Recommendation System

We often seek suggestions from friends, colleagues or known ones whenever we want to buy something like a refrigerator, TV, mobile phone or washing machine or even when planning for the Trip or which book to refer to or which movie or song for entertainment. Even with their best intentions, these friendly suggestions sometimes do not fit us or are effective in our case. Not just in decision making plays an imperative part to settle on choices which help to pick up benefits by connecting the best alternative as a suggestion. The point is that it is very arduous to highlight a precise suggestion on the items on which we might be interested.

One field of research in this direction is that of Recommender Systems. RSs are tools that use various techniques and algorithms to isolate irrelevant information from a huge amount of data and generate personalized suggestions of a small subset of them that a user can examine in a reasonable amount of time. An RS is an intelligent computer-based technique that predicts on the basis of users' adoption and usage and helps them to pick items from a vast pool of online stuff[15], Or it identifies the users' needs automatically by inferring the needs from the user's item interactions. Alternatively, the recommender system asks users to specify their needs by providing a list of keywords or through some other method[3]. RSs are a useful alternative to search algorithms since they help users to. These are the systems that help us to select similar things whenever we select something online. The concept of understanding a user's preference by their online behaviour, previous purchases, or history in the system is called a recommender system [12]

The recommender systems techniques can be used to address the problem of information overload by prioritizing the delivery of information for individual users based on user preferences. Recommender Systems are tools that use various techniques and algorithms to isolate irrelevant information from a huge amount of data and generate personalized suggestions of a small subset of them that a user can examine in a reasonable amount of time. So The task of the RS is to help the user to concentrate on the area of interest.

Following are the approaches of the job RS.

1. Collaborative Filtering recommenders
2. Content Based Filtering recommenders
3. Knowledge-based recommenders
4. Hybrid Recommenders

2.2.1 Collaborative Filtering Recommenders

CF uses similarity between users and items simultaneously to provide recommendations. CF RS finds users with similar interests as the target user and suggests recommendations to him/her based on their liked items. The key function in CF RS is the computation of similarities among users.[5]

As show in figure 2.1 as User P likes A, B and X the recommender system will try to recommend item X to the user Q

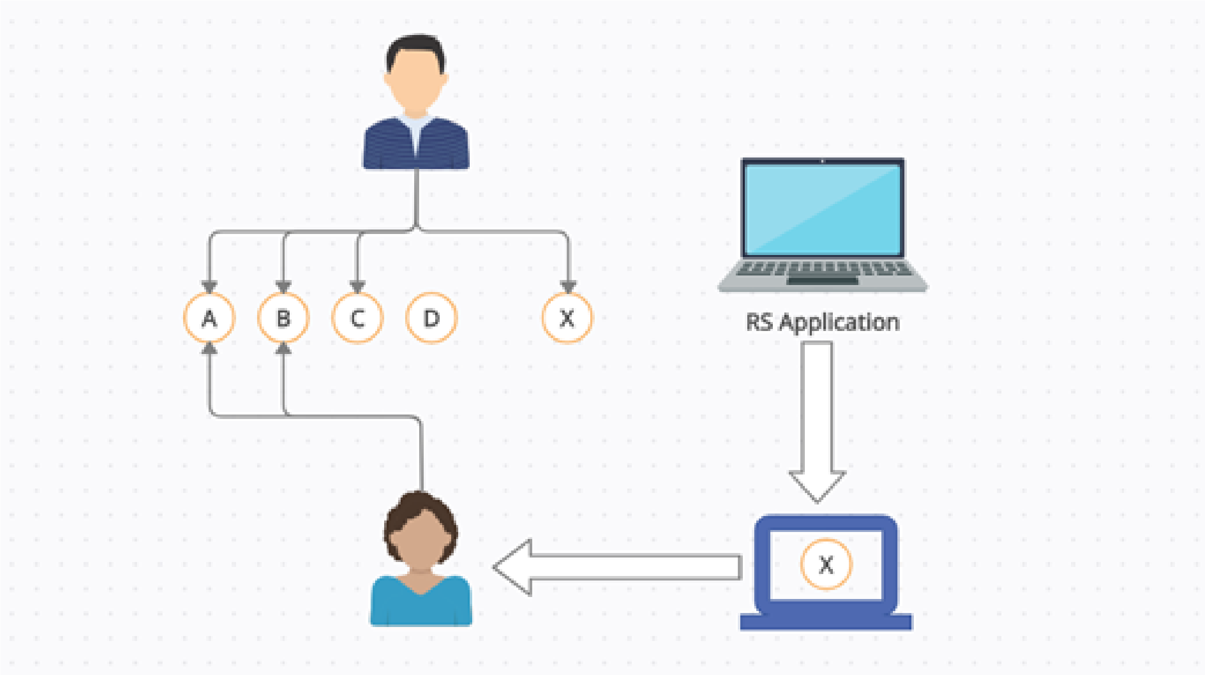


Figure 2.2: Collaborative Filtering System

Advantages

- CF RS does not require processing of the items so it is content independent.
- CF RS takes feedback from the users and the history of the users to recommend the items to the user.
- CF RS recommendations are based on user similarity.

Limitations

- A general problem of CF RS is the cold start problem, which may occur in three situations: new users, new items, and new communities or disciplines. If a new user rates few or no items, the system cannot find like-minded users and therefore cannot provide recommendations. If an item is new in the system and has not been rated yet by at least one user, it cannot be recommended. In a new community, no users have rated items, so no recommendations can be made and as a result.
- As it runs on the user's feedback, false feedback from the user can also cause the wrong recommendation
- Makes the criticism that CF systems are black boxes that cannot explain why an item is recommended except that other users liked it.

2.2.2 Content Based Filtering

CBF is based on a description of the item and a profile of the user's preferences. Items are recommended having similar content information to those a user has. CBF analyses the similar characteristics of the item and target users based on that build the profile for the user. In this system keywords are extracted from the item and user's description to find similarity between them. only the most descriptive features are used to model an item and users and these features are typically weighted. Once the most discriminative features are identified, they are stored, typically as a vector that contains the features and their weights. The user model typically consists of the features of a user's items. To find recommendations, the user model and recommendation candidates are compared in e.g. the vector space model To abstract the features of the items and user TF-IDF algorithm can be used to show similarity.[6]

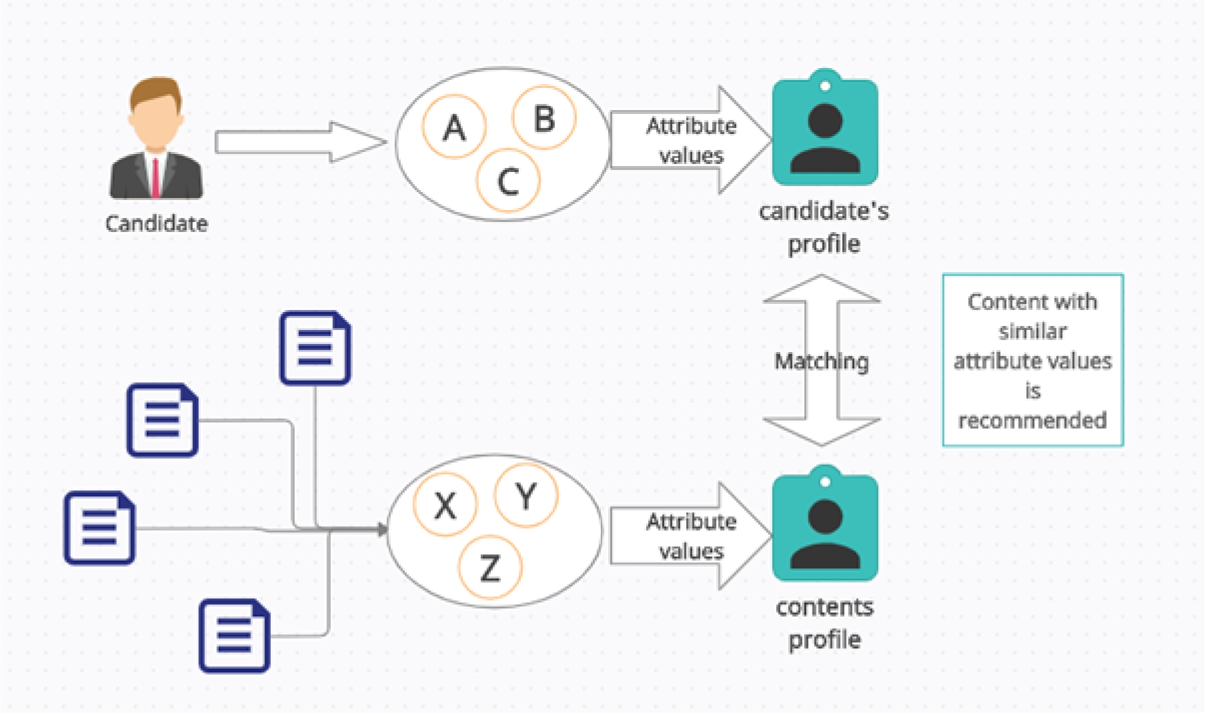


Figure 2.3: Content Based filtering System

As show in figure 2.2 it extracts attributes from the user and also from the job description later find similarity between them using various known technique.

Advantages

- The model doesn't need any data about other users, since the recommendations are specific to this user. This makes it easier to scale to a large number of users.
- The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.
- The user gets recommended the types of item they love.
- The user is satisfied by the type of recommendation
- New items can be recommended; just data for that item is required

Limitations

- The model can only make recommendations based on existing interests of the user. In other words, the model has limited ability to expand on the users' existing interests.
- The user will never be recommended for different items.
- Business cannot be expanded as the user does not try a different type of product.
- If the user matrix or item matrix is changed, the cosine similarity matrix needs to be calculated again.

2.2.3 Knowledge Based Recommender

To recommend the items which are less frequently used. In this technique, the relationship between user and item can be explicitly modelled. By using the knowledge of an item based on rules and patterns, we can recommend how a particular item is suitable for the user.[2]

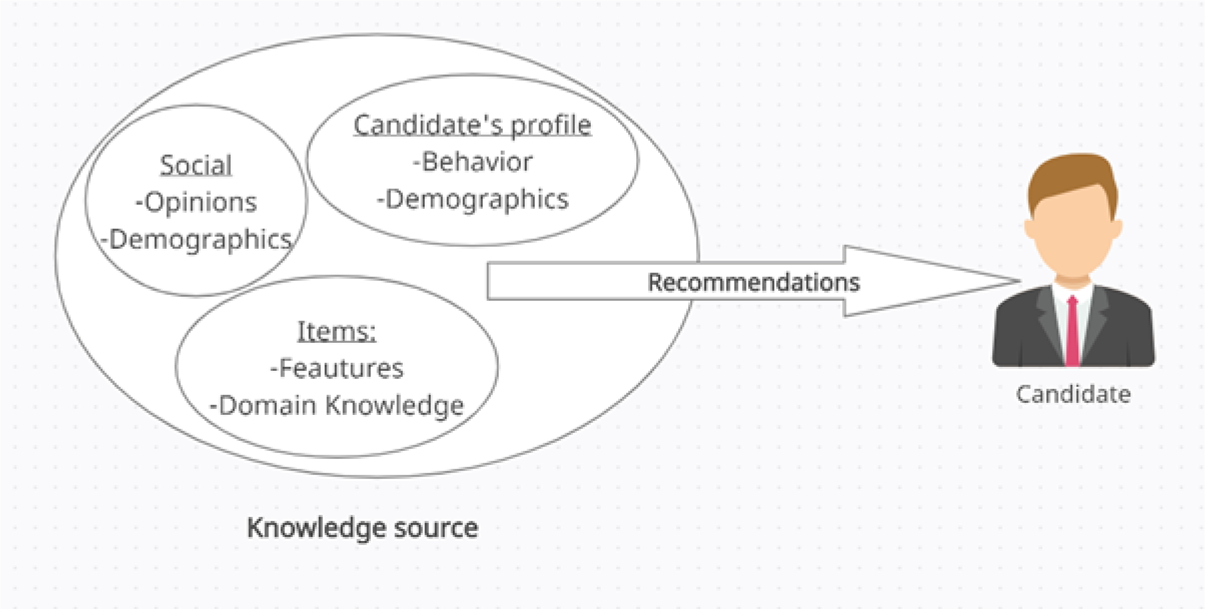


Figure 2.4: Knowledge Based Recommender System

Advantage

- It can recommend the new item to the user even when item is new in the system as it solves the problem of cold start

2.2.4 Hybrid Recommender System

Hybrid recommender technique is a mix of other techniques to override the drawback of the existing techniques.

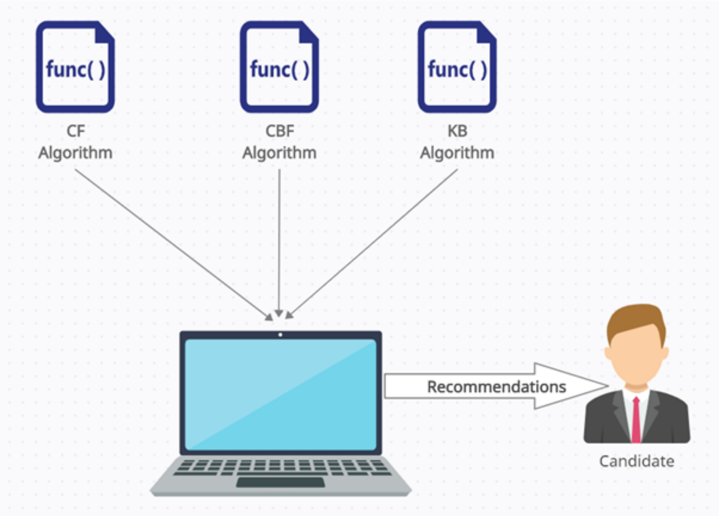


Figure 2.5: Hybrid Recommender System

As show in figure 2.4 Hybrid Recommender System takes input from the all suitable recommendation technique for the job recommendation problem.

2.3 Feature Extraction Using TF-IDF

Term Frequency – Inverse Document Frequency (TF-IDF) is an acronym for Term Frequency – Inverse Document Frequency. It is one of the fundamental approaches for representing a specific word or phrase to a given document in terms of information retrieval.

The TF-IDF believes that a document is nothing more than a ”bag of words.” TF just counts the number of times a word appears in a document. The document frequency (DF) is the number of times a word appears in a set of documents.

The TF-IDF value rises in proportion to the number of times a word appears in the text but is generally countered by the word’s frequency in the corpus, which helps compensate for the fact that some words appear more frequently than others general. Words having a high TF and DF may or may not be an essential metric for the paper. TF/DF, or Term Frequency * Inverse Document Frequency, is a measure of a word’s relation to a document. [7]

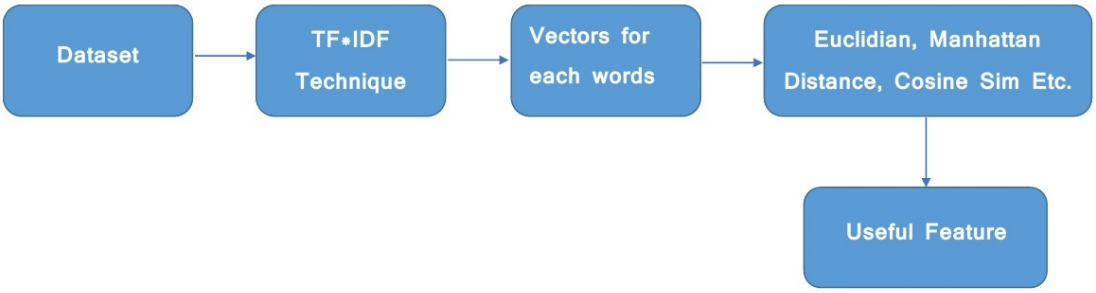


Figure 2.6: TF-IDF

2.4 Logistic regression

The method of modeling the probability of a discrete result given an input variable is known as logistic regression. The most frequent logistic regression models have a binary outcome, which might be true or false, yes or no, and so forth. Multinomial logistic regression is a type of logistic regression that can describe events with more than two distinct outcomes. Logistic regression is a useful analysis tool for determining if a fresh sample fits best into a category in classification tasks. Because our project requires the classification of documents into more than two categories, we will use multinomial logistic regression.

2.5 Random Forest

A random forest is a machine learning approach for classifying and predicting outcomes. It solves complicated issues using ensemble learning, combining multiple classifiers. A random forest algorithm is made up of several decision trees. The ’forest’ formed by the random forest technique is trained using bagging or bootstrap aggregation. Bagging is a meta-algorithm that combines machine learning approaches to improve their accuracy. The (random forest) algorithm decides the outcome based on decision tree predictions. By averaging or averaging the output of different trees, it predictions. As the number of trees increases, the precision of the output improves. The disadvantages of a decision tree algorithm are avoided by using a random forest technique. It enhances precision while reducing dataset overfitting.

2.6 Support Vector Machine

The Support Vector Machine (SVM) is a primarily supervised learning approach that may address both classification and regression problems. However, it is mainly used in Machine Learning to solve classification problems. The goal of the SVM method is to determine the best line or decision boundary for classifying n-dimensional space so that subsequent data points may be easily placed in the correct category. The best choice boundary is referred to as a hyperplane. SVM chooses the extreme points/vectors that help build the hyperplane. The approach is dubbed a SVM because support vectors reflect extreme cases.

2.7 KNN Algorithm

The Supervised Learning category includes the K Nearest Neighbor algorithm. The K-NN technique is extensively used for classification and regression. It is also a versatile algorithm for resampling datasets and imputing missing values. The name (K Nearest Neighbor) implies using K Nearest Neighbors (Data points) to anticipate the new Datapoint’s class or continuous value.

The algorithm’s learning process is as follows:
1. Instance-based learning: Rather than learning weights from training data to predict output (as in model-based algorithms), complete training instances are used to predict output for unknown data.

2. Lazy Learning: The model is not learned using training data before the prediction is required on the new instance, and the learning process is postponed until the prediction is asked.

2.8 Finding similarity

2.8.1 Cosine similarity

The similarity of two vectors in an inner product space is measured by cosine similarity. It detects if two vectors are pointing in the same general direction by measuring the cosine of the angle between them. In text analysis, it's frequently used to determine document similarity.
Let say There is two Vector A and B then we can find similarity between them using cosine

$$similarity = cos(\theta) = \frac{AB}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.1}$$

2.8.2 Euclidean distance and similarity

Let say There is two Vector A and B then we can find similarity between them using Euclidean distance

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{2.2}$$

Similarity can also be find using Euclidean distance using below formula

$$\frac{1}{1 + d(p,q)} \tag{2.3}$$

2.8.3 Supremum distance

The weighted euclidean distance can be calculated by assigning a weight to each property based on its perceived value.

$$d(i,j) = \sqrt{w_1|x_{i1} - x_{j1}|^2 + w_2|x_{i2} - x_{j2}|^2 + ... + w_m|x_{ip} - x_{jp}|^2} \tag{2.4}$$

Weighting can also be applied to ther distance measures as well.

Chapter 3

Literature Survey

This chapter briefly discusses the existing literature in the field of Job recommendation (JRD) system. Extracting the data from the resume of the user and job profile and relate them to show recommendations.

3.1 Overview of the recommendation system

Because of the Internet, companies have changed their hiring process by using the online platform. Companies choose to use online platforms because recruiting the appropriate person is a challenge faced by most companies. The unavailability of specific candidates in some skill areas has long been identified as a significant obstacle to the company's success

3.2 Boolean matching technique

Online channels like Internet job portals, social media applications, or a firm's career website have driven this development. While the companies established job positions on these portals, job-seekers use them to publish their profiles. For each posted job, thousands of resumes are received by companies. Consequently, a huge volume of job descriptions and candidate resumes are becoming available online. This vast volume of information gives a great opportunity for enhancing the matching quality; this potential is unused since search functionality in recruiting applications is mainly restricted to Boolean search methods. The need increases for applying the recommender system technologies that can help recruiters to handle this information efficiently.[8]

3.3 Context to recommend

We must consider unary attributes such as individual skills, mental abilities, and personality that control the fit between the individual and the tasks to be accomplished [14], as well as the relational attributes that determine the fit between the individual and the upcoming team members.

In this context, literature usually distinguishes between

1. person-job
2. person-team
3. person-organization fits

Many types of research have been conducted to discuss different issues related to the recruiting problem as well as the application of recommender system technologies. However, job recommendation is still a challenging domain and a growing area of research.[2]

Some of the followings are existing systems for a job recommendation.

- Hybrid job recommender System
  - A probabilistic hybrid approach
  - A proactive job recommender system
- Content-based job recommender systems
  - Machine learned recommender system

3.4 Hybrid job recommender systems

3.4.1 A probabilistic hybrid approach

The recommen-dation approach used both concepts: CBF and CF simultaneously. Its understands the individual preferences as a combination of preference factors. In a basic approach for CF, we look at each value of user/ object pairs (x, y), where x is a set of users and y is a set of objects. The model can then be represented as a variable z which is associated with each value of (x, y), assuming that x and y are independent conditioned on z. The model parameters are then estimated using the Expectation Maximization (EM) algorithm.[9]

This model produced a rating matrix that assigns assessed values to candidate□s profile containing the probability that recruiter x rates candidate y with value v. Later, they defined v = ”qualified”, ”not qualified”. Then, they transformed the rating matrix by replacing variable y with a variable a to represent the attributes that were extracted from the candidate resumes. As many attributes are assigned to several profiles, we will see the attribute several times with different values v.

To improve the match between people and jobs: a CV-recommender and a job recommender, separately. In the first step, they built a system recommending CVs that are similar to resumes previously selected by the recruiter for a specific job profile. In the second step, they developed a second RS that recommends jobs to candidates based on their preference profiles which are in turn based on previous preference ratings.[9]

Limitation

- It answers in binary only either 0 or 1 cannot answer in rank wise to give recommendations.

3.4.2 A proactive job recommender system

The proactive recommender system is an adaptive system that attempts to integrate the idea of recommender systems.[13] This system contains five components: web spider, ontology checker, profile analyzer, preference analyzer, and user interface generator. Web spider is a parser that periodically acquires job information from an exterior source. The ontology checker matches information with ontologies and performs the classification. Then, the job data is stored in a pre-designated form. The profile analyzer makes the recommendations whenever the users modify the group of favorites by comparing the weight differences with current open jobs. Then, a list of recommended jobs is generated. Finally, the preference analyzer deduces the explicitly defined user’s preferences and gives a recommendation for preferred jobs after calculating the similarity of jobs to the user’s preference.[4]

Limitations

- One way recommendation only recommend to the job seeker
- Cold start problem as user change profile

3.5 Content-based job recommender systems

3.5.1 Machine learned recommender system

The recommendation problem is treated as a supervised machine learning problem. They build an automated system that can recommend jobs to applicants based on their past job histories, in order to facilitate the process of choosing a new job. An item in this learning model represents a person who is hired in an organization. Each item is characterized by a set of features extracted from the candidate’s resumes. Given a person who is currently working in an organization, they want to predict the next organization. If the accuracy of such predictions is sufficiently high, the model can be used to recommend organizations to employees who are seeking jobs. This approach uses all past job transitions as well as the data of both employees and organizations to predict an employee’s next job transition. They train a machine learning model using a large number of job transitions extracted from person profiles available on the web.[11]

Limitations

- As it takes previous or historic data into the consideration, the problem of sparsity and cold start could occur.

Chapter 4

Proposed Work

The Project aims to analyze the information from the resume of the candidate and the job description posted by the employer/organizer and extract the useful information from the resume and job description to find the similarity. For that, job descriptions are classified into several categories to identify the role of the job. Then Based on the similarity, the model can recommend the job to the candidate or it helps the organizer/employer for shorting the candidate in the initial phase.

4.1 Logical Development

Finding the relevant job based on the candidate profile from the large amount of information present on the internet is time consuming and cumbersome. Also for particular jobs posted on the internet, a massive number of applications are received to the employer which makes it hard to go through each resume manually. Thus, an effective technique is required to recommend an appropriate job for the candidate and to shortlist the most suitable resumes for an employer. For that purpose the job descriptions are classified into several categories using the classifier model such as Logistic Regression, Random forest and Support vector machine. Using the above three models one having better accuracy will be chosen as model. Then the features from the resume like tools/technologies, work experience, location info , educational background and expertise are used to find or recommend the suitable and appropriate job to the candidate over the pool of jobs. Jobs are recommended using by finding the similarity of the documents which can be calculated using the cosine similarity or kNN to recommend the top-n matching items.

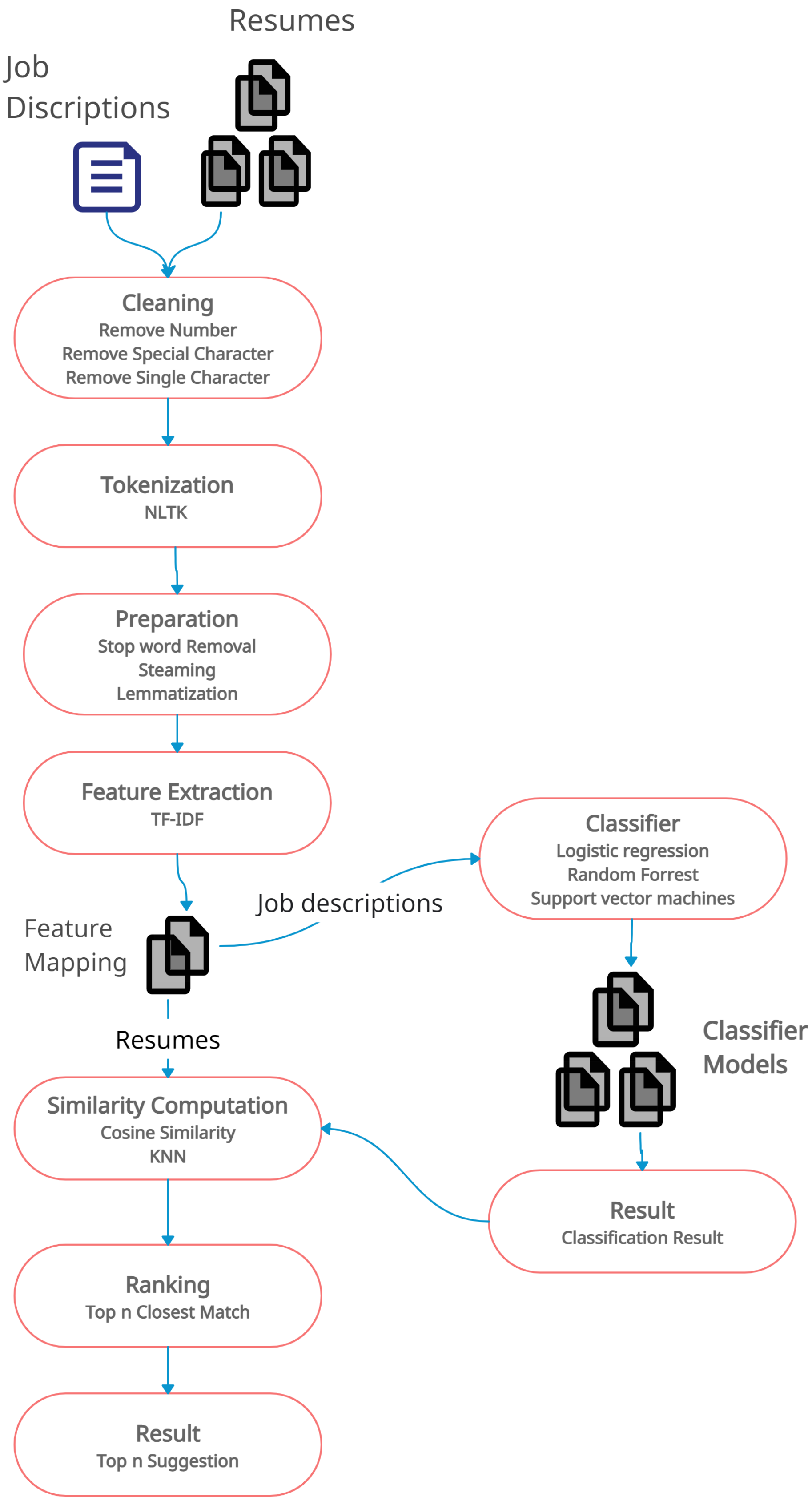


Figure 4.1: Block diagram describing the proposed methodology

4.2 Proposed Methodology

A Systematic approach is proposed in order to achieve the objectives mentioned in the above Figure 4.1.The proposed work consists of the four mother phases. The first phase is Data extraction and Data preprocessing. The next phase is Feature Extraction where various features will be extracted from the resume and the job description which are useful to finding or recommending a job. The third phase is classification of the job description in which the classifiers will be trained to classify the job description into the various categories as per the job profile or position role. The final step is about computing the similarity between the documents to recommend.

4.2.1 Data Extraction and Data Preprocessing

The resume is in the PDF file format in which the related words and text is in a blocked structure not as the simple line. To extract the text which is related to its context the tika algorithm is used which identifies the block in the pdf structure and extracts the information plain text. Here pdf is read which may contain the noise, or undesired character or spacing. For that data preprocessing happens.

Data preprocessing[1] is one of the most required steps in data analysis in order to achieve maximum accuracy and throughput . It includes techniques to remove incomplete data, making data consistent and ready to use for experiments. Mostly, a library called pandas[10] is used for such preprocessing. Preprocessing text involves converting it into a form that is predictable and analyzable for the task. Data cleaning, data transformation, and data reduction are procedures involved in data preprocessing.

Data Cleaning involves handling of missing data, noisy data. Strategies to handle missing data involve removing the tuples, filling the missing values. In noisy data Lowercasing, Stemming, Lemmatization, Stop words removal such as ‘a’, ‘.’, ‘,’, ‘an’, ‘the’, removing extra spaces, new line and the unknown character from the description such as ‘ ’, ‘ª’, ‘º’, ‘¿’ outlier analysis can be done to clean irregular and inconsistent data such as experience of the candidate. Lowercasing is one of the simplest and most effective forms of text preprocessing.

Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens. The Resume usually doesn’t consist of the whole english sentence, it only consists of the words so the resume can be tokenized as words using NLTK. While the Job description consists of full english sentences which can be done using tool gensim.

Stemming is the process of reducing inflection in words (e.g. experienced ,experience) to their root form (e.g. experience). The ”root” in this case may not be a real root word, but just a canonical form of the original word. NLTK provides the implementation of stemming. Stemming is desirable as it may reduce redundancy as most of the time the word stem and their derived words mean the same. Lemmatization is very similar to stemming, where the main aim is to remove inflections and map a word to its root form. The only difference is that lemmatization tries to do it properly. It doesn't just remove things off, it links words with similar meaning to one word.

4.2.2 Feature Extraction

To extract the features from the resume such as tools, languages, work experience, projects model can use a bag of words which match with the heading of each block described in block structure of the resume.

- **Tools:** tools, IDE, editors
- **Work experience:** work experience, past history, job role, achievement
- **Projects:** projects, publication, certificate

From the job description useful features to recommend jobs such as required skills, minimum years of experience, any brownie or plus point and job location can be useful which can be extracted using

- **Required skills:** required skills, Technical skills, soft skills
- **Plus point:** plus point, brownie point, good to have, advantageous

And from that TF-IDF methods can be used to extract the essential feature from this preprocessed text. TF-IDF is a useful method here, a resume mostly in size of 1 or 2 pages and same in case of job description too. Normalization of the words will be used to handle the varying frequency of the words in the document.

4.2.3 Classification of job description

According to the body of the text, job descriptions can be classified into clusters, which facilitates the model in identifying the position of the job advertised by the organization/employer. As an example,

- Backend developer
- UI/UX developer
- Data Scientist
- ML/AI
- Cloud Architect

The classification of these job descriptions can be done using the classification methods such as Support vector machine (svm), Logistic regression or Random forest. The model having better results can be chosen for the subsequent process of recommendation. Here svm can find the optimum line which separates the job description into different profiles.

This process is useful to overcome the computation overhead. Job descriptions are clustered based on the class so a new job description can be classified and for a particular resume we don't need to go through all the job description to recommend the job we can use representative or representatives of the cluster to match on the initial stage.

4.2.4 Computing similarity

In case of bag of words or vector of words, cosine similarity between the vector of the job description and the resume can be computed to find likeness and k-nearest neighbour to compute the distance in case of clusters. Clusters having less distance can be used to recommend. Different distance measures can be useful to find the distance such as euclidean distance, Manhattan distance or supremum distance. Supremum distance is useful in case the different features have different weights which can affect the decision of the recommendation system.

4.3 Summary

This project, Thus systematically presented an overview of the methodology proposed. By dividing our problem into parts such as classification and the similarity computation, we intend to modularize our problem solving approach.

Chapter 5

Simulation

We are going to use the dataset from the dice website which is having 21999 rows and 12 attributes which is having job description, employment type, company name, job location, post date and required skills. This dataset is useful as a data set of job descriptions. Other dataset of stackoverflow which has 64460 rows and 61 attributes which has features such as database skill, Education level, job factors, experience which can be used as a user database. By combining these both datasets our dataset is created.

This chapter discusses the visualisation of the data set for. It consists of bar graph for given skill in the market for a job-seeker.

5.1 visualisation

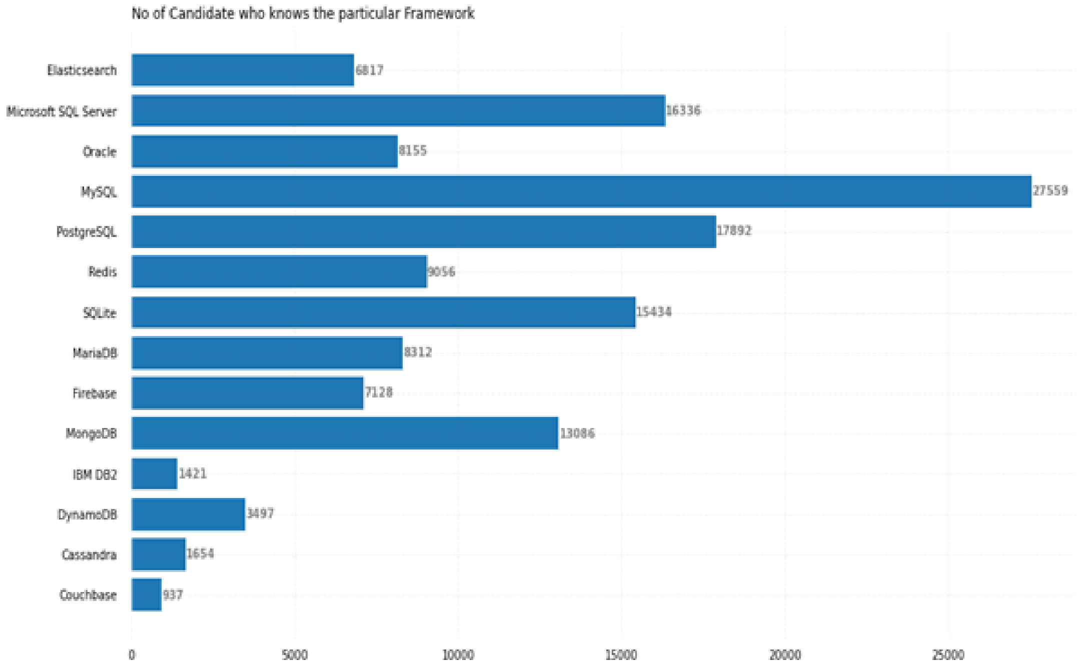


Figure 5.1: Show number of users know particular Language

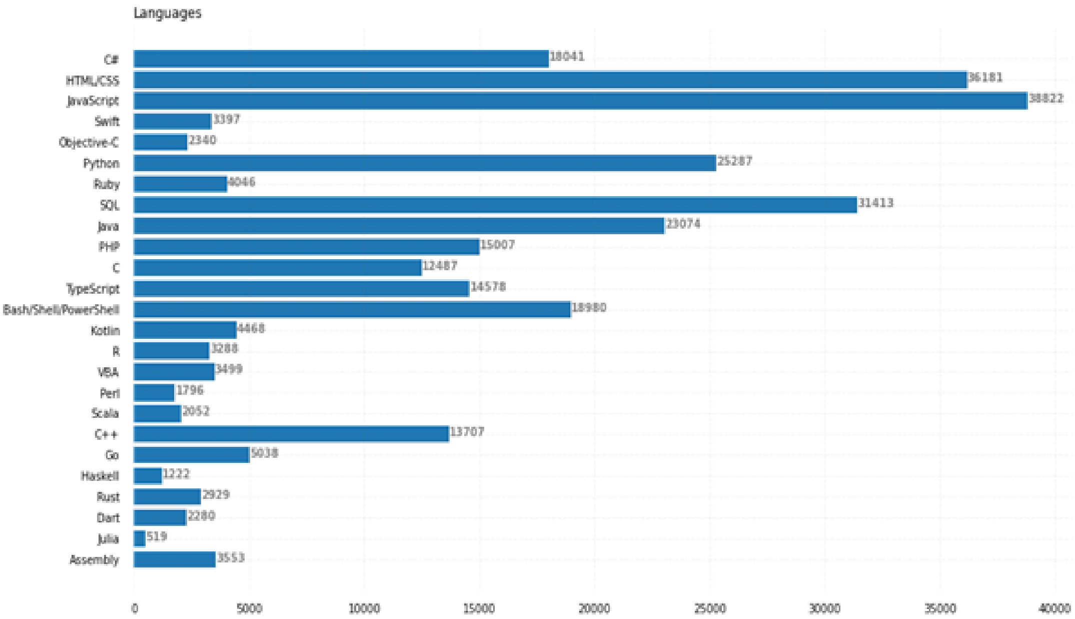


Figure 5.2: Show number of users familiar with particular Database

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Talent acquisition is most import task for the success of the company. In current situation for a given job thousands of job seeker apply which make hard for the hiring team to go through each and every resume manually and check for the credibility of the applicant. Similar for the job seeker in the large market thousands of the jobs are available which makes finding suitable job difficult for the user. This project aims to solve this problem by making automation of the resume matching process by using various technique for the data extraction from the given text or description and finding similarity between the job seeker's profile and job description.

6.2 Future works

In this report we have currently described the present literature survey and the technique which can be used to recommend the job. After deciding which technique is better, we can implement the suitable best technique or the mix of techniques. Also keywords and data extraction can also be performed in future to gather data and build profile for user or job.

References

[1] Suad A Alasadi and Wesam S Bhaya. “Review of data preprocessing techniques in data mining”. In: *Journal of Engineering and Applied Sciences* 12.16 (2017), pp. 4102–4107.

[2] Shaha Alotaibi. “A survey of job recommender systems”. In: *International Journal of the Physical Sciences* 7 (July 2012). DOI: `10.5897/IJPS12.482`.

[3] Joeran Beel et al. “Research-Paper Recommender Systems: A Literature Survey”. In: *Int. J. Digit. Libr.* 17.4 (Nov. 2016), pp. 305–338. ISSN: 1432-5012. DOI: `10.1007/s00799-015-0156-0`. URL: `https://doi.org/10.1007/s00799-015-0156-0`.

[4] P. Brusilovsky and D. H. Lee. “Fighting Information Overflow with Personalized Comprehensive Information Access: A Proactive Job Recommender”. In: *Autonomic and Autonomous Systems, International Conference on*. Los Alamitos, CA, USA: IEEE Computer Society, June 2007, p. 21. DOI: `10.1109/CONIELECOMP.2007.76`.

[5] Google Developer. *Collaborative Filtering and Matrix Factorization*. URL: `https://developers.google.com/machine-learning/recommendation`. (accessed: 23.09.2021).

[6] Google Developer. *Content based filtering*. URL: `https://developers.google.com/machine-learning/recommendation`. (accessed: 23.09.2021).

[7] Ankur Dhuria. *How to do content based filtering using TF-IDF?* URL: `https://medium.com/analytics-vidhya/how-to-do-a-content-based-filtering-using-tf-idf-f623487ed0fd`. (accessed: 23.09.2021).

[8] Tim; Faerber Frank; Weitzel and Tobias Keim. “”An Automated Recommendation Approach to Selection in Personnel Recruitment”. In: 2003.

[9] Thomas Hofmann and Jan Puzicha. “Latent Class Models for Collaborative Filtering”. In: IJCAI’99. Stockholm, Sweden: Morgan Kaufmann Publishers Inc., 1999, pp. 688–693.

[10] Wes McKinney et al. “pandas: a foundational Python library for data analysis and statistics”. In: *Python for high performance and scientific computing* 14.9 (2011), pp. 1–9.

[11] Ioannis K. Paparrizos, Berkant Barla Cambazoglu, and Aristides Gionis. “Machine learned job recommendation”. In: *RecSys ’11*. 2011.

[12] Baptisite Rocca. *Introduction to recommender System*. URL: https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada. (accessed: 23.09.2021).

[13] J. Ben Schafer, Joseph Konstan, and John Riedl. “Recommender Systems in E-Commerce”. In: *Proceedings of the 1st ACM Conference on Electronic Commerce*. EC ’99. Denver, Colorado, USA: Association for Computing Machinery, 1999, pp. 158–166. ISBN: 1581131763. DOI: 10.1145/336992.337035.

[14] Tomoki Sekiguchi. “Person-Organization Fit and Person-Job Fit in Employee Selection: A Review of the Literature”. In: *Osaka Keidai Ronshu* 54 (Jan. 2004).

[15] Pradeep Singh et al. “Recommender Systems: An Overview, Research Trends, and Future Directions”. In: *International Journal of Business and Systems Research* 15 (Jan. 2021), pp. 14–52.

Acknowledgement

We would like to express our deep gratitude to our project guide, Dr. Sankita J Patel, Associate Professor, Computer Engineering Department, SVNIT Surat, for their valuable guidance, helpful feedback, and co-operation with a kind and encouraging at the initial stage. We would also like to thank Dr. Mukesh A. Zaveri, Head of Department, Computer Engineering Department. We are also thankful to SVNIT Surat and its staff for providing this opportunity which helping us to gain sufficient knowledge to make our work successful. Special thanks and appreciation to our colleagues in developing and people who have willingly helped us out with their abilities