Towards partial fulfilment for Undergraduate Degree Level Programme
Bachelor of Technology in Computer Engineering

A third stage project Evaluation Report on:

Job Recommendation System

Prepared by:

Admission No.	Student Name
U18CO009	Tarsariya Keyur
U18CO010	Jadhav Ganesh
U18CO018	Shekhaliya Shubham
U18CO039	Nainuji Jigar

Class : B.TECH. IV (Computer Engineering) 8th Semester

Year : 2021-2022

Guided by : Dr. Sankita J. Patel

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY, SURAT
- 395 007 (GUJARAT, INDIA

Student Declaration

This is to certify that the work described in this project report has been actually carried out and implementedby our project team consisting of

Sr.	Admission No.	Student Name
1	U18CO009	Tarsariya Keyur
2	U18CO010	Jadhav Ganesh
3	U18CO018	Shekhaliya Shubham
4	U18CO039	Nainuji Jigar

Neither the source code there in, nor the content of the project report have been copied or downloaded fromany other source. We understand that our result grades would be revoked if later it is found to be so.

Signature of the Students:

Sr.	Student Name	Signature of the Student
1	Tarsariya Keyur	
2	Jadhav Ganesh	
3	Shekhaliya Shubham	
4	Nainuji Jigar	

Certificate

This is to certify that the project report entitled Job Recommendation System is prepared and presented by

Sr.	Admission No.	Student Name
1.	U18CO009	Tarsariya Keyur
2.	U18CO010	Jadhav Ganesh
3.	U18CO018	Shekhaliya Shubham
4.	U18CO039	Nainuji Jigar

Final Year of Computer Engineering and their work is satisfactory.

SIGNATURE:

GUIDE JURY HEAD OF DEPT.

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. The Doc2Vec embedding is used to learn the features from the highly unstructured text data. 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. Rather than calculating similarity between all the documents the resumes are clustered using centrality based clustering algorithms and job descriptions are classified. Later for computing the similarity only representative or representatives of the groups are used.

Keywords: Recommendation System - POS tagging - Job recommendation System - Content based Filtering - Doc2Vec - Clustering - Classification - 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

This project aims to recommend relevant resumes and jobs for particular job titles and resumes respectively from large data. The project uses Doc2Vec embedding technique for vectorising the job description titles and the cosine similarity measure is used to find similarities between two vectorised judgements.
However, vectorising a large volume of job description titles using Doc2Vec poses scalability issues. This project performs the following in order to achieve these objectives.

1. Resume Cluster Preparation: To overcome the scalability issues, k-means clustering is performed on the resumes features.

2. Training Doc2Vec embedding on each job description title: Document Embedding and a bag of the word is trained on each Job description title using various features like skills and job description words or tokens.

3. Job Description classification: job description classification Using various models like Logistic Regression, Support Vector Machine One Vs Rest Classifier and Naïve Bayes classifier etc.

Finally, this project implements a job recommendation That will recommend top n job for each queried resume and top n resumes for each queried job description.

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 6 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.

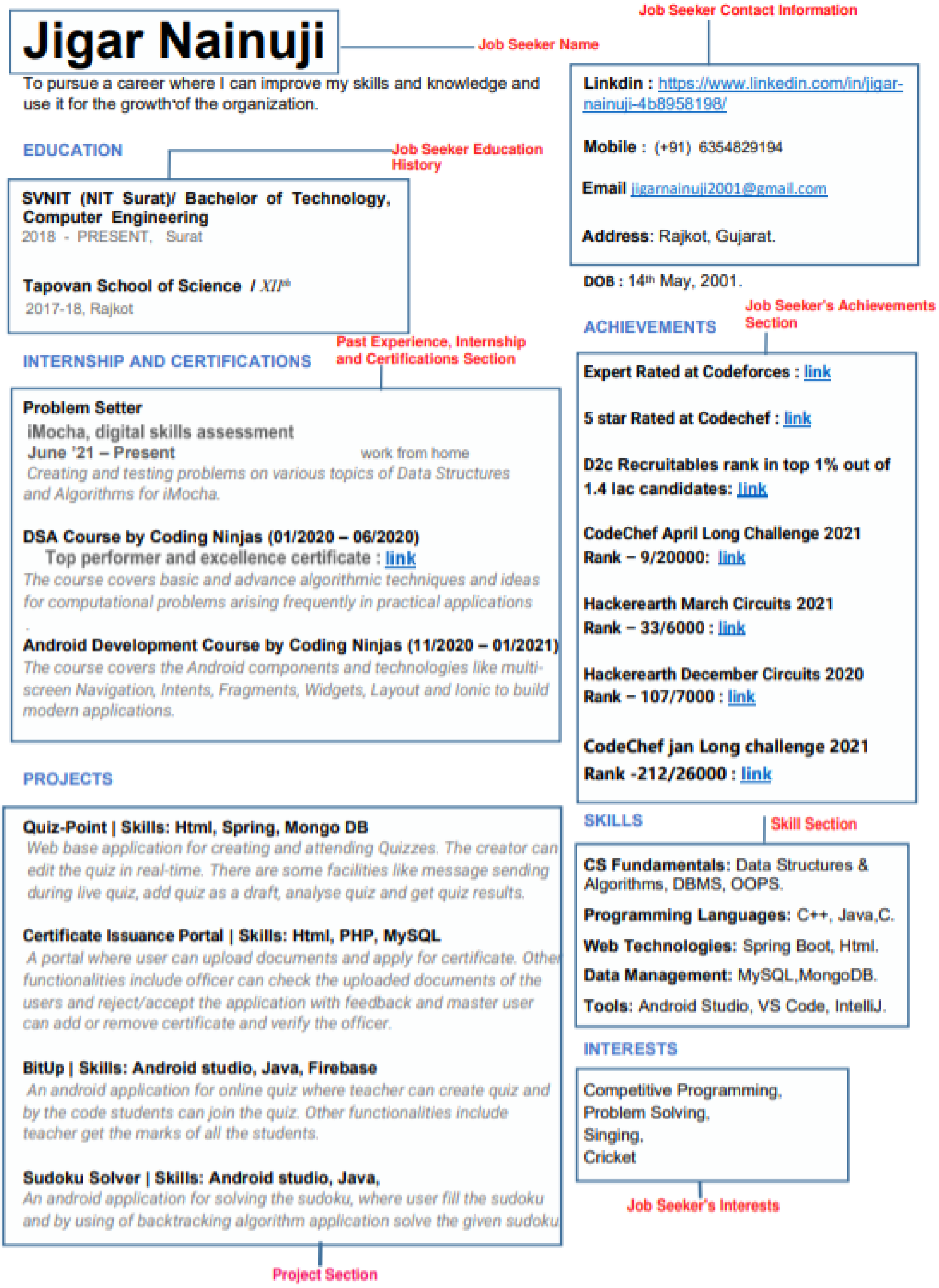
2.1 Resume

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.



Job Seeker Name

Job Seeker Contact Information

Job Seeker Education History

Past Experience, Internship and Certifications Section

Job Seeker's Achievements Section

Skill Section

Job Seeker's Interests

Project Section

Jigar Nainuji

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

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

Mobile : (+91) 6354829194

Email jigarnainuji2001@gmail.com

Address: Rajkot, Gujarat.

DOB : 14th May, 2001.

EDUCATION

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

Tapovan School of Science / XIIth
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.

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

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

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.

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.

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 Job Description

The Job description provides a high-level overview of the role, level, and scope of responsibility. A job description is a document containing information about the job title, job purpose, required qualification, required minimum experience, knowledge, skills, and abilities where experience Identifies the minimum number of full-time experience required in terms of years and the type of work experience that an employee needs to be qualified for the job.

2.3 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[18], 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 [15]

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

2.3.1 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]

2.3.2 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.

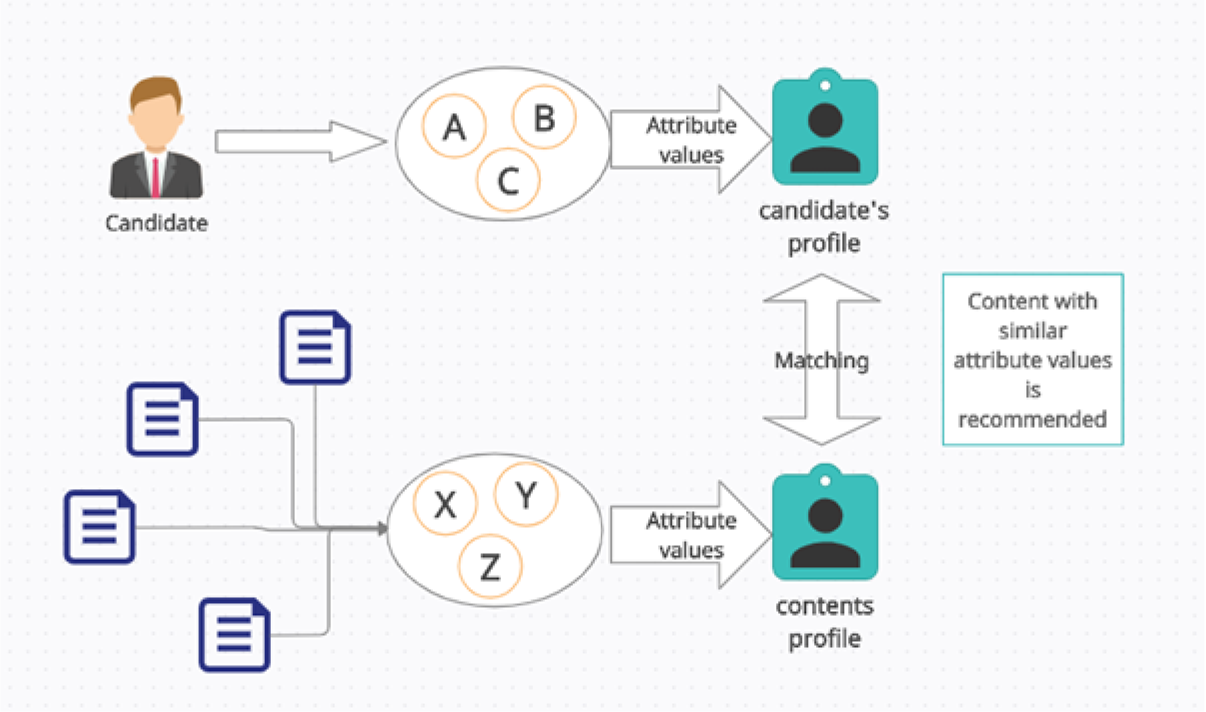


Figure 2.2: Content Based filtering System

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.
- 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.
- Business cannot be expanded as the user does not try a different type of product.

2.3.3 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]

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

2.3.4 Hybrid Recommender System

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

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

2.4 Doc2Vec

Doc2Vec provides a technique for extracting word embedding from corpus paragraphs. Instead of vector representations of the entire corpus, we can think of these word vectors as paragraph vectors. Doc2Vec modelling is a technique for learning text vector representations.[8] Vectorization is also applied to short portions of text documents, ranging from phrases or sentences to huge documents. This technique can be used to anticipate words in paragraphs. These models can predict a word in a particular context by concatenating the section with numerous word vectors from a paragraph.

Using the word vectors as a starting point for learning is a good concept. Even though we can predict the next word in the sentence using a random factorization word vector, we can also expect the next word using the paragraph vector, which contains information about the semantic relationship and can help us get better results.

Doc2Vec models have two variants :

- Distributed Memory Model
- Distributed Bag Of Words

The continuous Bag Of Word Models is similar to the Distributed Memory Model, while distributed bag of words is alike to skip gram model.

2.4.1 Distributed Memory Model (DMM)

Distributed Machine Learning is a multi-node Machine Learning system that improves performance, increases accuracy, and scales to larger input data sizes. It reduces errors made by the machine and assists individuals in making informed decisions and analyses from large amounts of data. Distributed machine learning algorithms have evolved to handle enormous data sets. Distributed ML algorithms are integral to large-scale learning because of their ability to allocate learning processes onto several workstations to enable faster learning algorithms.

2.4.2 Distributed Bag-of-Words (DBOW)

The Distributed Bag-Of-Words (DBOW) model aids in determining the context words associated with a target word. The allocated memory model approximates the word using the context of surrounding words, whereas the dispersed bag of words model approximates the word using the context of surrounding words. The target word is used in the dispersed bag of words model to match the word's context. When comparing a distributed bag of words with skip-gram models, remember that the skip-gram model takes the target word as input. We find that the distributed bag-of-words models outperform the distributed memory models in several comparisons.

2.5 Classification :

2.5.1 Support Vector Machine Algorithm

SVM stands for Support Vector Machine and is one of the most widely used Supervised Learning algorithms for classification and regression tasks [13]. The purpose of the SVM method is to find the best line or decision boundary for categorising n-dimensional space into classes so that fresh data points can be placed in the correct category rapidly in the future. This best decision boundary is called a hyperplane. The extreme points(vectors) that assist create the hyperplane are chosen via the Support Vector Machine. Support vectors describe these severe circumstances, and the process is known as a Support Vector Machine.

Linear Support Vector Machine: Linear Support Vector Machine is used for linearly separable data, which means that if a dataset can be classified into two groups using a single straight line, it is linearly separable data, and the classifier employed is called Linear Support Vector Machine.

2.5.2 Naive Bayes

Naive Bayes is a classification technique established on Bayes' Theorem and the presumption of predictor independence [14]. A Naive Bayes classifier, in simple terms, asserts that the existence of one part in a class is irrelevant to the presence of any other feature. The Naive Bayes model is simple and useful for large data sets. Naive Bayes outperforms even the most advanced classification systems due to its simplicity.

2.5.3 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.

2.6 Clustering

Clustering, often known as cluster analysis, is a machine learning technique for classifying unlabeled data. It can be defined as follows: ”A method of sorting data points into different clusters based on their similarity. The objects that have possible similarities are kept in a group that has few or no similarities to another group.”

2.6.1 K-Means Clustering Algorithm

K-Means Clustering is an iterative and Unsupervised Learning algorithm that groups the unlabeled dataset into various clusters in such a manner that each dataset belongs to only one group that has similar properties. [11] It permits us to cluster data into various groups and supplies a straightforward method to specify the classifications of groups in an unlabeled dataset without any training. It’s a centroid-based procedure, which means that each cluster has its centroid. The primary goal of this technique is to reduce the sum of distances between data points and the clusters that they belong to.

The k-means clustering algorithm primarily executes two tasks:

- An iterative process determines the best value for K centre points or centroids.
- Allots each data point to its nearest k-centre. Those data points which are near to the respective k-centre, make a cluster.

2.6.2 Elbow Method

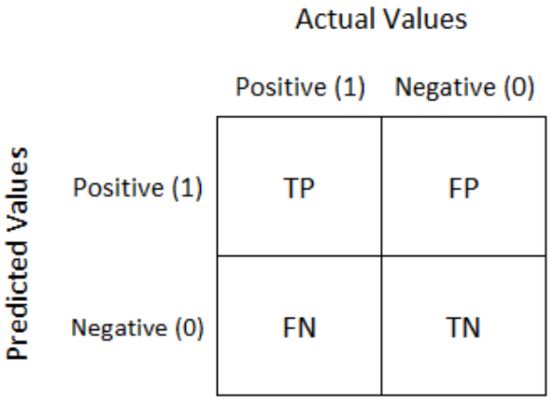
The Elbow technique is one of the most favoured ways to find the optimal clusters. This technique uses the idea of WCSS value. WCSS stands for Within Cluster Sum of Squares, representing the total variations. [9]

The formula to calculate the value of WCSS (for 3 clusters) is given below:

$$WCSS = \sum_{PiinCluster1} distance(P_i, C_1) + \sum_{PiinCluster2} distance(P_i, C_2) + \sum_{PiinCluster3} distance(P_i, C_3) \tag{2.1}$$

2.7 Performance measures for classification problem

Confusion Matrix is an evaluation matrix used to evaluate a prediction model's performance by comparing the actual values to the predicted values. It has 4 parameters that are used to calculate different measures: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN).



		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Figure 2.3: Confusion Matrix

Accuracy:

Accuracy is the ratio of correct predictions to the total number of predictions. The accuracy evaluation matrix is observed to be a good evaluation metric when a balanced data set was considered.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.2}$$

Precision:

Precision is the ratio of correct positive predictions to the total positive predictions. Higher precision indicates that false positives are fewer.

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

Recall:

Recall is the ratio of correct positive predictions to the total actual true values.Higher recall indicates that false negatives are fewer.

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

F1 Score:
F1 score is harmonic mean. F1 score is the mean of precision and recall, which aims to generalise precision and recall metrics. This evaluation measure has been taken into consideration since the testing dataset is imbalanced.

$$F1{-}Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.5}$$

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

$$similarity = cos(\theta) = \frac{AB}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.6}$$

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

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{2.7}$$

Similarity can also be find using Euclidean distance using below formula

$$\frac{1}{1 + d(p,q)} \tag{2.8}$$

2.8.3 Supremum distance

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

$$d(i, j) = \sqrt{w_1|x_{i1} - x_{j1}|^2 + w_2|x_{i2} - x_{j2}|^2 + ... + w_m|x_{ip} - x_{jp}|^2} \tag{2.9}$$

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.[6]

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 [17], 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.[7]

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.[7]

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.[16] 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.[12]

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 and resumes clustered to improve similarity finding efficiency. 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 a 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, Naïve Bayes and Support vector machine. Using the above three models one having better accuracy will be chosen as a model. Then the features from the resume like tools/technologies, work experience, location info, educational background and expertise are used to find or recommend a 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. Also before finding similarities resumes are clustered using a k-mean clustering algorithm to improve model efficiency.

4.2 Proposed Methodology

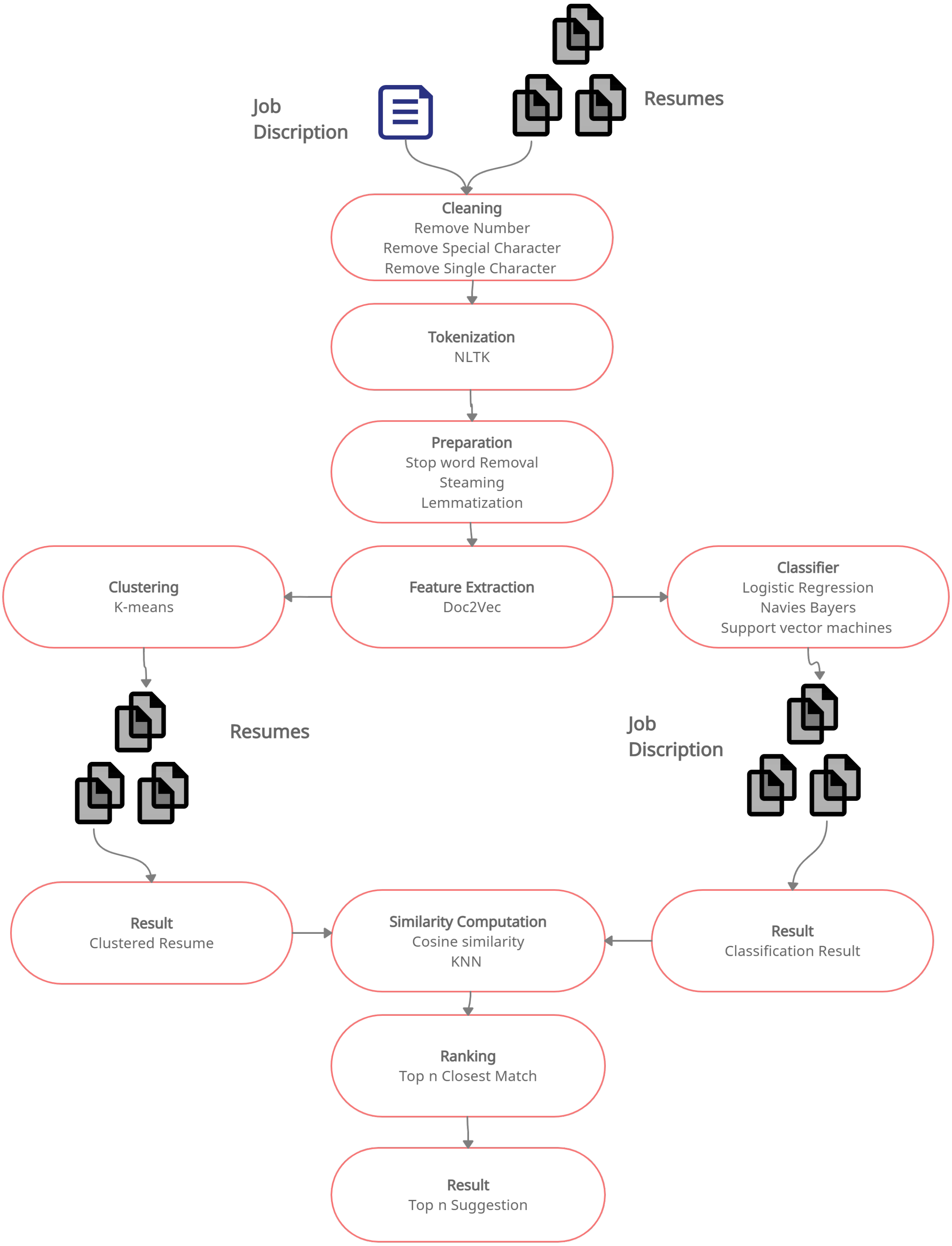


Figure 4.1: Block diagram describing the proposed methodology

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 a 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 the 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 and similar resumes are clustering into similar types of groups. 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

4.2.3 Classification of job description

According to the body of the text, job descriptions can be classified into classes, 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 naive Bayes. The model having better results can be chosen for the subsequent process of recommendation.

This process is useful to overcome the computation overhead. Job descriptions are classification 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 descriptions to recommend the job we can use representative or representatives of the class to match on the initial stage.

4.2.4 Clustering of Resume

According to the body of the resumes, skills and experience of candidate resumes, 's can be clustered into various groups of similar resumes, which help at the time of finding similar resume's for particular jobs.
The clustering of resumes can be done using the clustering method k-means. This process is useful to overcome the computation overhead. Resumes are clustered based on the body of resumes so a new resume can be clustered and for the particular Job descriptions, we don't need to go through all the resumes to recommend the resumes we can use cluster representatives to match on the initial stage.

4.2.5 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 And Results

This chapter describes our experimental simulations and results. It consists of a description of our initial data preprocessing techniques, along with the description and testing results of the various algorithms utilized for performing classification and clustering of the documents. It also discusses the selection of optimal hyperparameters for the doc2vec model. Later, the clustering is performed on the resumes to cluster similar types of the resume using the K-means algorithms and classification of the job descriptions done using the various algorithms using Logistic regression and SVM with linear kernel, and their corresponding results are discussed. Finally, this chapter presents the performance obtained by training Doc2Vec to embed different hyper-parameters to obtain their optimal values.

5.1 Data Extraction and Pre-Processing

The experiments have been conducted using the dataset of the dice website, which is the leading American website for IT jobs. This dataset contains the job descriptions of the 22000 and the title of the jobs. For the resumes dataset, the data from the Kaggle was used, which contains the resume's text. The dataset of job descriptions contains lots of variants in the job title name, so to overcome this problem, various proper job titles are listed, and as a part of data preparation, the similarity between job titles calculated using the cosine similarity one having result more than the threshold value is selected for the model training After this we remained with around 6000 data rows.

In the working prototype of this experiment, the user is supposed to upload a job description and resume in the pdf document file. The tika tool and pdfminer are used to extract the data from the pdf document. The characteristic of the tika tool is that it extracts the information block-wise from the pdf so that valuable and related information remains together. At the same time, pdfminer is useful to extract the information line by line. Extracting the data from the pdf files may contain noise such as non-ASCII keywords. For that, Traditional text preprocessing techniques are used to remove the inconsistency and non-uniformity in the text. The following traditional preprocessing steps are applied to highly unstructured data.

- Transforming the text content into the lower case
- Individual words are extracted and tokenized with the help of white spaces and line breaks
- stemming and lemmatization are performed to convert the given word into a root word
- Elimination of white spaces, newline and punctuation marks.
- Removal of stop-words such as ”a”, ”an”, ”the” etc.
- Removal of non-ASCII keywords

5.2 Extracting data from Document

This section defines the various techniques utilised for extracting the needed information such as the name, email, phone numbers, experience, or skills from the document file. The regex string matching method extracts the emails and phone numbers from the document file by scanning the entire Document after preprocessing the data.

The POS tagging method is used to extract the candidate’s name and experience. After Pos tagging on the text, each token is categorised in the verb, adverb, noun Etc., based on the corpus context. In the resume document, generally, the candidate name comes as two consecutive nouns. This is implemented using the nltk library. The regex grammar for it prepared, and it passed to the chunking by traversing all possible chunked tree name can be found. To extract the experience from the text is tokenized by the new line character and tokenized by line. An important observation is that lines containing the experience keyword have a cardinal token. This way can find experience from the document text. The same way can find brownie point as well as a minimum qualification.

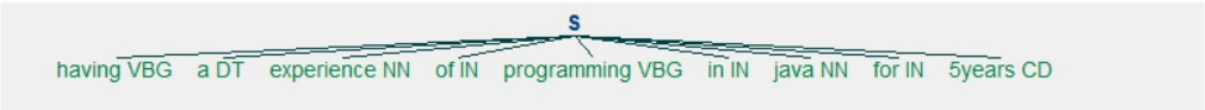


Figure 5.1: Chunk

The skills database has been prepared from the aggregated dataset of the StackOverflow in which each word is tokenized using the comma, space and slash. With the help of 2-gram and

3-gram on each token, the dataset for the keywords of the skills is prepared. 2-gram and 3-gram are generated and matched with the dataset to extract the skills from the document file again. Furthermore, the final list of skills is prepared.

5.3 Experiment setup

The project has three significant motives. The first is to perform the clustering on the text document corpus of the resume to cluster similar resumes. The second is to perform the classification on the text document corpus of the job descriptions dataset. The Third is to train doc2vec embedding on each document to obtain the optimum hyperparameters. The experiments are performed on a machine with Intel Core i5-8250U processor running at 1.60 GHz using 8 Gb of Ram with 240 Gb of SSD, running on windows 10.

5.4 Cluster Formation Result Analysis

This section discusses the various experimentation carried out and presents the results obtained in cluster formation. The final clusters formed after this act as represent documents. Here the dataset of the results is unlabeled. To group the similar documents we need unsupervised learning algorithm.

5.4.1 Clustering Result Analysis

K-Means clustering is performed on the resume dataset with doc2vec embedded model and different value of the k that is number of cluster is determined by the elbow method. In which the graph agains the squared cost and different values of k is plotted. The point at which we got sudden drop in squared cost that is finilised as the number of clusters. Before passing document in doc2vec model the data cleaning and preprocessing is done on the dataset. As the K-Means is highly affected by the noise.

For the K-Means algorithm doc2vec model trained for the parameter as vector size = 400, window = 10 and epoch=10. Then with the 600 iteration the algorithm run for different values of k from 1 to 15 and elbow method used to get optimum value of the k.

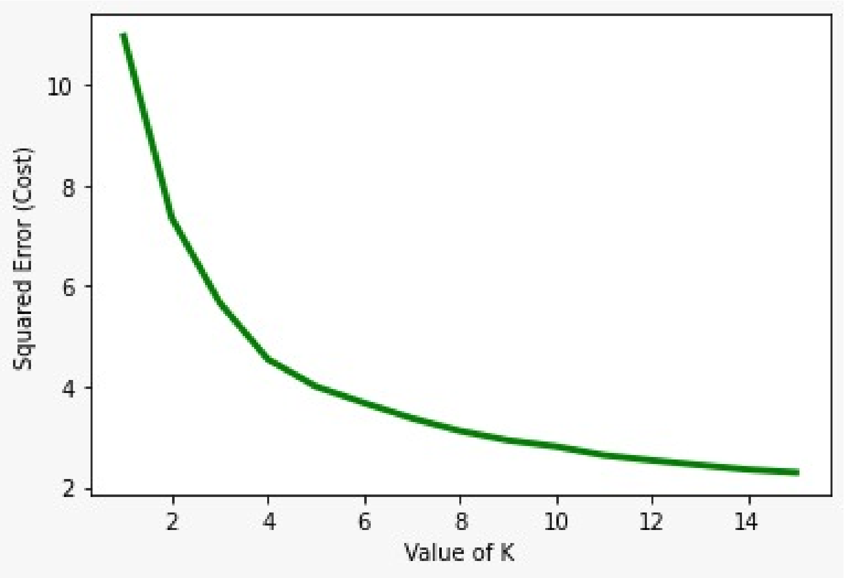


Figure 5.2: Squared Error vs K

As shown in Figure 5.2 value of squared error minimize for the value of K = 10.

This is cluster formation of the dataset of the resumes having 10 clusters the squared error cost for the above is 1.77.

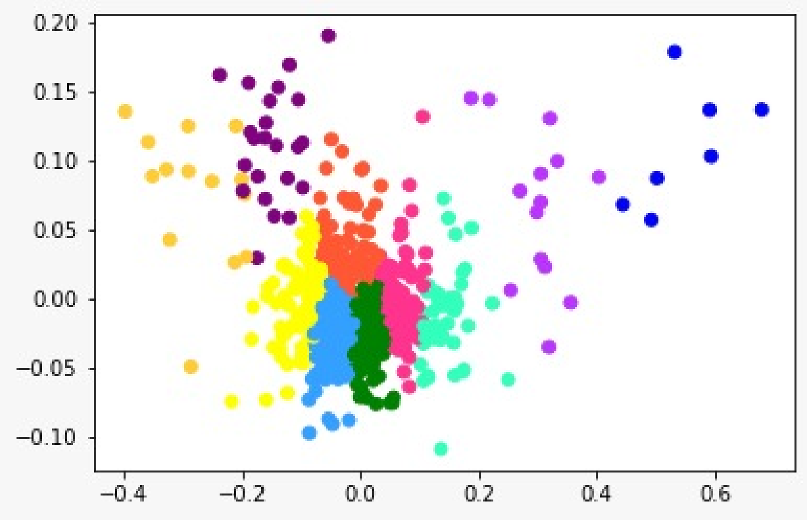


Figure 5.3: Clustered Resumes

5.5 Classification Result Analysis

This section discusses the various experimentation carried out and presents the results obtained in classification formation. After this, similar documents are classified into the corresponding group. The job descriptions dataset was labeled.

The Problem of the job description is the Problem of multiclass classification. For that, LogisticRegression, Support vector classifier and naive Bayes classifier are tested for the trained model from the doc2vec.

Classification Method	**Accuracy**	**F1 score**
logistic regression	0.7266	0.7166
svm linear kernel	0.6626	0.6590
Naïve Bayes	0.7373	0.7386

As shown in above table naive bayes classifier gives highest accuracy out of all the given classification.

5.6 Doc2Vec Model Training Result Analysis

Since this project focuses on recommending similar resumes to the queried job description so After applying preprocessing on job description text each of these job description's text must be vectorised for efficient processing. Thus, this project conducts various experiments for finding optimal hyperparameters to train the Doc2Vec model. Once each job description is vectorised, a similar resume to the job description can be then found based on the cosine similarity of vectorised documents.

5.6.1 Hyperparameters for training Doc2Vec

The hyperparameters of training Doc2Vec model for which optimal values need to find out are:

Vector size (vSize): dimensionality of the feature vectors

Window (w): the maximum distance between the current and predicted word within a sentence.

Epochs (iter): Number of iterations (epochs) over the corpus.

For evaluating the performance of Doc2Vec models on our dataset Accuracy and F1 Score have been utilised as the evaluation metrics First, Vector Size is varied, keeping the window fixed at 10 and epochs set to 30. As shown in Figure 5.3 and Figure 5.4, the highest accuracy of 0.7386 and an F1 - Score of 0.7246 is obtained for Vector Size = 325; thus, it is selected.

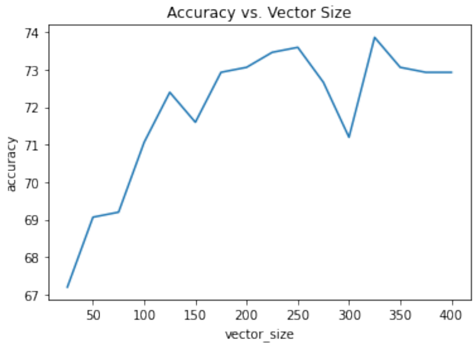


Figure 5.4: Accuracy vs. Vector Size

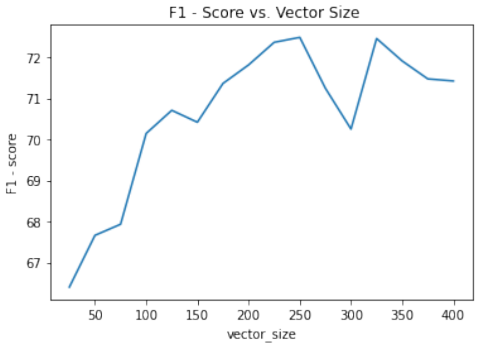


Figure 5.5: F1 - Score vs. Vector Size

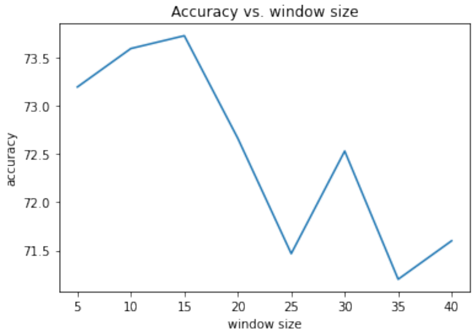


Figure 5.6: Accuracy vs. Window

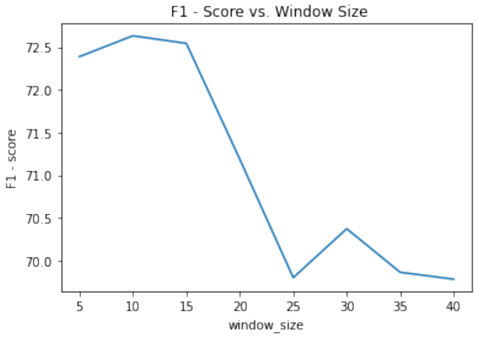


Figure 5.7: F1 - Score vs. Window

Now, the window values are varied, keeping the Vector Size fixed at 325 and epochs set to 30. As it can be seen from Figure 5.5 and Figure 5.6, a Window = 15 gives the highest accuracy of 0.7373 and an F1 - score of 0.7254; thus, it is selected.

Finally, epochs were varied, keeping the Vector Size fixed at 325 and Window fixed at 15.

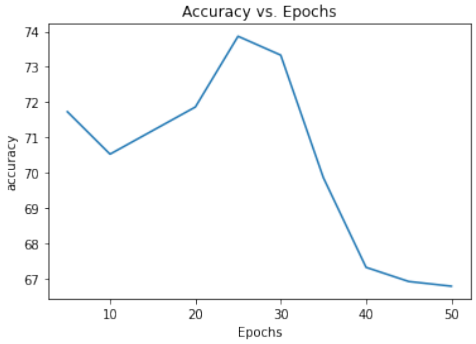


Figure 5.8: Accuracy vs. Epochs

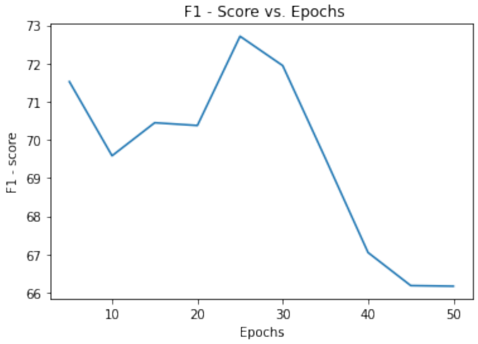


Figure 5.9: F1 - Score vs. Epochs

It is evident from Figure 5.7 and Figure 5.8 that epochs = 25 gives the highest accuracy of 0.7386 and an F1 score of 0.7271; thus, it is selected.

Hence, from the various experiments performed, Vector Size = 325, Window = 15 and epochs = 25 are obtained as the best optimal hyperparameters for training the Doc2Vec model.

5.7 Conclusion

This chapter presents the implementation of the objectives of the project. The project's first task is to cluster the resumes. K-Means clustering is performed to achieve this objective, and the elbow method is used to determine the optimum value of the k.

The next task is to about the document classification of the job descriptions. Three different models, SVM, logistic regression, and Naive Bayes classifier, are used for classification. The Naive Bayes classifier gives promising results. The dataset job description contains the candidate's required skills, which are independent features. In this point of view, the Naive Bayes treats them as independent and gives the best result. The next major task is to select hyperparameters optimally for the doc2vec model.

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

The work proposed in this project can be extended by considering all the job types. Later for the purpose of the 2-way recommendation system, For finding match of the document it is matched with the representatives of the cluster or class one having higher similarity is considered for the match. The purpose of grouping them in class or cluster is to save computational time of the system for every input. Later for extracting the information from the document it is scanned multiple time from the algorithm to overcome this issue multithreading environment can be used.

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

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

[8] Donghwa Kim et al. “Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec”. In: *Information Sciences* 477 (2019), pp. 15–29. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2018.10.006. URL: https://www.sciencedirect.com/science/article/pii/S0020025518308028.

[9] Dhendra Marutho et al. “The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News”. In: *2018 International Seminar on Application for Technology of Information and Communication*. 2018, pp. 533–538.

[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] Shi Na, Liu Xumin, and Guan Yong. "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm". In: *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*. 2010, pp. 63–67.

[12] Ioannis K. Paparrizos, Berkant Barla Cambazoglu, and Aristides Gionis. "Machine learned job recommendation". In: *RecSys '11*. 2011.

[13] Ashis Pradhan. "SUPPORT VECTOR MACHINE-A Survey". In.

[14] Irina Rish. "An Empirical Study of the Naïve Bayes Classifier". In: *IJCAI 2001 Work Empir Methods Artif Intell* 3 (Jan. 2001).

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

[16] 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`.

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

[18] 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 science Engineering Department, SVNIT Surat, for their valuable guidance, help- ful feedback, and co-operation with a kind and encouraging at the initial stage. We would also like to thank Dr. Mukesh A. Zaveri, Professor, Computer science 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.