Journal of Information Systems Engineering and Management
2025, 10(5s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Intelligent Job Recommendation System based on Semantic Embeddings and Machine Learning

Priyanka Singla, Vishal Verma
Ch. Ranbir Singh University
Jind, India.
singla.priyanka198@gmail.com, vishal.verma@crsu.ac.in

ARTICLE INFO

Received: 07 Oct 2024

Revised: 15 Dec 2024

Accepted: 26 Dec 2024

ABSTRACT

To address the shortcomings in existing approaches of job recommendation systems, this paper proposes a novel machine-learning-based job recommendation system that performs bi-directional matching for dynamic and accurate recommendations. The proposed approach generates ideal job recommendations for a targeted Curriculum Vitae (CV) and vice versa. Unlike previous approaches, the proposed approach incorporates natural language processing (NLP) techniques to extract linguistic features such as Bag of Words (BoW), n-grams, TF-IDF, and Parts-of-Speech (PoS) tag and build a rich feature set. These features are further analyzed using semantic embeddings, enabling robust job matching. Experiments were performed to validate the performance of the proposed approach. The designed system is validated on various real-world datasets, overcoming the dataset size limitations of prior works. Due to combination of semantic embeddings, machine learning, and various similarity measures, this approach demonstrates the potential to deliver reliable, explainable, and ideal job recommendations, addressing the challenges of static and false outputs in existing systems.

Keywords: Hybrid Job Recommendation System, Similarity Index, Clustering, Job Matching, CVs, Semantic Embeddings, NLP Techniques , SBERT

1. INTRODUCTION

Job recommendation systems have become quite popular in recent years. The existing job recommendation systems can fall in any of the five categories such as 1) content-based recommendation systems, 2) knowledge-based systems, 3) collaborative-based systems, 4) Reciprocity-based systems, and 5) Hybrid-based systems. Each of these types of recommendation systems have their own pros and cons and have peculiar features [1-3]. The existing methods used for transforming data from CVs and job descriptions into a structured format are error-prone due to ambiguity and can result in loss of information. Previous studies have experimented on just small datasets. Moreover, no existing study has used a hybrid approach that used semantic embeddings, machine learning (ML) and various similarity measures for job matching and recommendation.

The existing recommendation systems exhibit various shortcomings including false and static recommendations. False job recommendations can be defined as job recommendations that are not according to the preferences, skills or career aspirations of a user [4]. Similarly, static job recommendations can be defined as job recommendations that are not up to date rather to job market or the user behaviour over-time [5]. To find a novel and significantly improved solution for job recommendations, the key features of the existing systems were identified. The existing classical recommendation systems are used for transforming data from CVs and job descriptions into a structured format that can be error-prone and can result in loss of information [6-7]. The existing works are experimented with and validated on 1 to 2 datasets, while the proposed approach is tested with 4 real-world datasets. Moreover, the existing studies mainly bank on hybrid approach that use machine-learning and simple frequency measuring techniques like TF-IDF, similarity measures for job matching and recommendation [8].

The proposed approach aims to solve the problem of job recommendations; a machine learning-based solution is discussed in this paper. The proposed solution will perform bi-directional matching for job recommendations. The bi-directional job recommendation was previously achieved in [9-10]. However, [9-10] has just used content-based filtering on random features that can occasionally result in static and false recommendations. There is a need for a job recommendation system that is robust and efficient and can perform dynamic and true job recommendations. However, the proposed approach works in two phases. Modern approaches like LLM (Large Language Models) [11,12] are also being used but these approaches are not explainable. The proposed work is based on NLP (Natural Language Processing) based techniques to extract the various linguistic features (such as Bag of Words (BoW), n-grams, TF-IDF, and Parts-of-Speech (PoS) tags from the data. In the second phase, the extracted features are analyzed using semantic embeddings and then machine learning is used for recommending relevant jobs. A novelty of the proposed job-recommendation systems is that it shall provide true job recommendations.

The rest of the paper is structured into a set of sections. Section 2 describes the outcomes of the detailed literature survey conducted to find out the research gap in the field of job recommendation systems. Section 3 presents the methodology of improved and enhanced job recommendation using Sentence-BERT and Cosine Similarity. Section 4 presents the experiments details and results of the experiments. The paper is concluded with Section 5.

2. RELATED WORK

A detailed literature survey was conducted to find out the existing contributions in the field of job recommendation systems. The key aims of this literature review-based study was to investigate the types of methods that are typically employed for job recommendations, finding the common methods/techniques leveraged in implementation of modern job recommendation systems, and identifying the key challenges existing in the field of job recommendation that require more attention. In search of these aspects, a detailed research survey was conducted from all major research databases and 339 studies were found. All these studies were carefully analyzed and only 38 relevant and peer-reviewed studies were selected. After detailed study of these 38 studies seven key challenges were identified around Job recommendation systems. A few of these identified challenges are cold-start problem, fairness and bias mitigation, explainability and transparency, skills and competency mapping, context-aware recommendations, reciprocity, and psychological factors as discussed in Table 1.

Table 1: Comparison of general work with existing similar job recommendation studies

Citation	**Year**	**Work**	**Type**	Used Dataset	**Used features**	**Used Methodology**
[1]	2017	Job Matching	Content-based recommendation	From Jan 2010 to July 2016 all Profiles registered having evaluation from "Most High (12,823)" to "Most Low (2,644)").	• Job Features: workplace nearest station, start date, salary, position and tasks, requisite skills • Candidate Features: personal information, skill and qualification, assessment results, work history, and previous working history with staffing company child care/elderly care priorities.	• Used text mining • Keywords are extracted by using KH Coder. • To identify meaningful keywords, experts were interviewed • Identified a number of important keywords indicating positive or negative influences on job matching • frequency list of keywords • hierarchical cluster analysis • co-occurrence network
[2]	2017	Job Recommendation	Reciprocity-based recommendation	129 Chinese employers recruited postgraduate students of Beihang University	• Time started hunting job • Time to get a job offer • Workplace nearest station • GPA, • Student vote	• create students' profiles • student similarity calculation • job recommendation using weights

				from 2012-15.	• Employer information	
[3]	2017	Job Recommendation	Combining content-based and collaborative filtering	Not Mentioned	• Job Features: Job class, job skills, JobReqYear • Candidate Features: Class, city, skill, degree, most recent job title, last company, previous applied job	• Used Relational functional gradient boosting to learn features • Cost-sensitive learning with RFGB • Built a regression tree for
[4]	2022	Job Suitability Measurement and Prediction	Content-based Recommendation	CRs and JDs dataset	• Job Features: Job title. City, description, responsibilities, Education, Required skills • Candidate Features: Title, city, description, work experience, Education, Skills, certificates	• Linear Regression • Decision Tree • Adaboost • XGBoost
[5]	2022	Job Recommendation	Content-based Recommendation	CRs and JDs dataset	• Education • Experience • Skill • Traits	• Competitive analysis • Personality trait analysis • Apply the DISC model • Term-frequency matrix • Similarity calculation • Job Recommendation
[6]	2016	Personalized Job Matching System	Content-based Recommendation	A small dataset 10K jobs	• Degree • Domain-specific skills • College • Discipline	• Parsing and Tokenization • Semantic Labelling • Pattern matching • Measure skills similarity • Constructing an ontology of skills
[7]	2007	Matching Human Resources	Collaborative-based Recommendation	Not Mentioned	CV Data	• multilayer framework for matching collaboration partners • recommender-based approach to matching human resources
[8]	2020	Job Recommendtion	Hybrid-based Recomme	The dataset includes job offers and job-seeker	• Job offers data • Job seekers interaction data • ratings, likes, and	• Data Collection • Textual processing of job offers • Joc clustering

			ndation	interactions such as ratings, likes, and reviews scrapped from web	reviews	• Matching and rating • Job recommendation
	2024	**Proposed Work**	Hybrid Approach (Semantic Embeddings + ML)	real-world dataset of 10K jobs Source: Kaggle	• Job Features: Workplace (city and country), Nearest station, Working Mode, Start date, Salary Position, Job Role and Duties requisite skills, offer details. • Candidate Features: Personal information, Skill Qualification, Experience Assessment results, Work history	An intelligent approach is used: 1) uses SBERT to extract features and uses machine learning to predict matching job 2) uses cosine similarity to extract lexical features and uses ML for classification

In the last two decades, a lot of research contributions have been made in the field of job recommender systems to address key challenges. However, after reviewing the major contributions from the year 2015 to 2024, it has been identified that the research area of job recommender systems continues to face several open research challenges [13-21]. Moreover, there is a pressing need that experts, researchers and analysts to actively explore the methods and techniques that help in improving the fairness, effectiveness, usability and adaptability of the job recommender systems.

3. METHODOLOGY

This section presents the proposed approach for dynamic and true job recommendations. The proposed approach is divided into two sections: the first section prepares the dataset, and the second section analyzes the data to provide true job recommendations. For data preparation, steps like data selection and data integration were used to prepare datasets. For data analysis, both user's data sets (CVs) and job description datasets were given input. Then various linguistic features (such as Bag of Words (BoW), n-grams, TF-IDF, and Parts-of-Speech (PoS) tags) were extracted from the data. For the linguistic feature extraction, LLM (Large Language Models) and NLP (Natural Language Processing) NLP techniques were used. The purpose of using two different techniques for feature extraction was the validation of the process of feature extraction. Afterwards, the extracted features were passed on to the machine learning models such as Random Forest (RF) and Logistic Regression (LR) and deep learning models such as BERT and LASTM were used for job recommendations. The process of the proposed approach is shown in Figure 1.

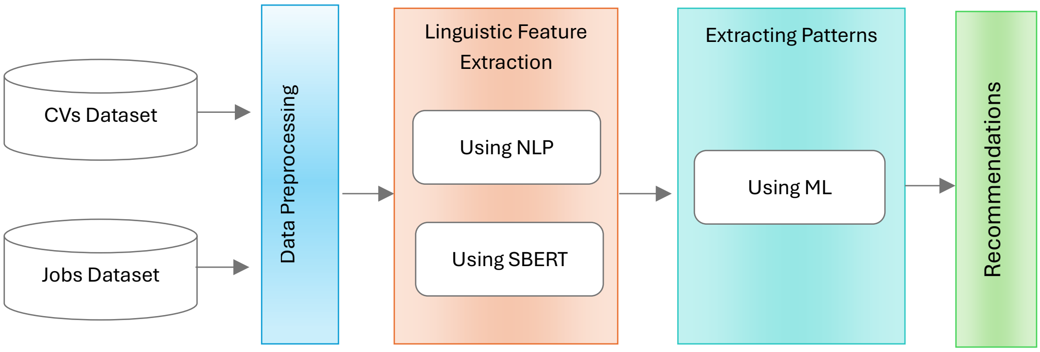

Figure 1- Machine Learning and Deep learning-based True Job Recommendation

The fundamental principle of finding jobs opportunities is that Job opportunities are retrieved by similarity measures among the skill sets given in resume by the job seekers and Jobs descriptions given in advertisement by the job's recruiters. The proposed model gets the input of Jobs Descriptions (J) and Resume (R) as document. These Jobs Descriptions (J) and Resume (R) as documents are preprocessed by using the text cleaning, tokenization, stop words removal and lowercasing. These two preprocessed documents are given to two BERT models that are designed to generate the embedding of semantically meaningful document embedding. That is the reason SBERT is used in the proposed model. This model has the capability to understand the context and semantics of the word of the sentence because the SBERT has trained on large carpus that is the preseason this is called pre-trained model. Fine tuning the BERT modeled as SBERT to generate the embedding as fixed size vector representation of entire sentence or document. The semantic meaning of the text was captured by embedding. After that the token embedding generated by the BERT, Mean Pooling is applied by the fine tune S-BERT(Sentence-BERT) to aggregate the token embedding into single vector for document representation. Here, the average of all tokens embeddings is computed by Mean Pooling to result in a fixed size vector. These documents embeddings are used to measure the similarity between the documents. The similarity is measured by cosine angle between vectors. The proposed job recommender system recommends the Top-n jobs based on similarity score.

The following is the explanation of various steps of the proposed model. Input parameters are as below:

R: Set of CVs $\{R_1, R_2, \ldots\ldots R_n\}$

J: Set of Job Descriptions $\{J_1, J_2, \ldots\ldots J_n\}$

N: Number of top recommended jobs for each CV.

Algorithm for Job Recommendation System Using K-Means and SBERT

The following is the designed algorithms that uses CVs and job descriptions data and generates Top-N job recommendations against the given CV.

Input: Collect datasets R and J containing CVs and job descriptions, respectively.

Step-1 Split text data into individual tokens (words) of both datasets as below:

$$R_i = Tokenize(R_i) \qquad \forall R_i \in R$$

$$J_i = Tokenize(J_i) \qquad \forall J_i \in J$$

Step-2 Convert all tokens to lowercase as below:

$$R_i = Lowercase(R_i) \qquad \forall R_i \in R$$

$$J_i = Lowercase(J_i) \qquad \forall J_i \in J$$

Step-3 Remove commonly used words that do not add significant meaning as below:

$$R_i = R_i - StopWords \qquad \forall R_i \in R$$

$$J_i = J_i - StopWords \qquad \forall J_i \in J$$

Step-4 Construct the Bag-of-Words representation for both datasets as below:

$$BoW(R_i) = \{word1 : freq1, word2 : freq2, \ldots..\}$$

$$BoW(J_i) = \{word1 : freq1, word2 : freq2, \ldots..\}$$

Step-5 Apply SBERT to transform textual data into dense vector representations E_R and E_J

$$E_{R_i} = SBERT(R_i) \qquad \forall R_i \in R$$

$$E_{J_i} = SBERT(J_i) \qquad \forall J_i \in J$$

Step-6 Generate SBERT embeddings for each CV against all job descriptions

$$E_{R_i,J_i} = SBERT(R_i, J_i) \qquad \forall R_i \in R, \forall J_i \in J$$

Step-7 Compute cosine similarity for each CV embedding E_{R_i} and each job description embedding E_{J_i}, as below:

$$CosineSimilarity(R_i, J_i) = \frac{E_{R_i} \cdot E_{J_i}}{\|E_{R_i}\| \times \|E_{J_i}\|}$$

Step-8 Construct a cosine similarity matrix S, where each entry $S_{i,j}$ represents similarity between R_i and J_i. Apply K-Means clustering to group similar job descriptions for each input CV (R_i).

$$Clusters = KMeans(S, k)$$

Step-9 For each CV R_i, sort the job descriptions by cosine similarity and recommend the top N jobs.

$$TopN(R_i) = Classify\left(Cosine_{Similarity_{score}for} R_i, descending\right) [:N]$$

Figure 2 shows the recommendation model that employs machine learning with S-BERT model and measures the similarity among the job seeker according to their skills sets and jobs description in advertisement produced by job recruiters.

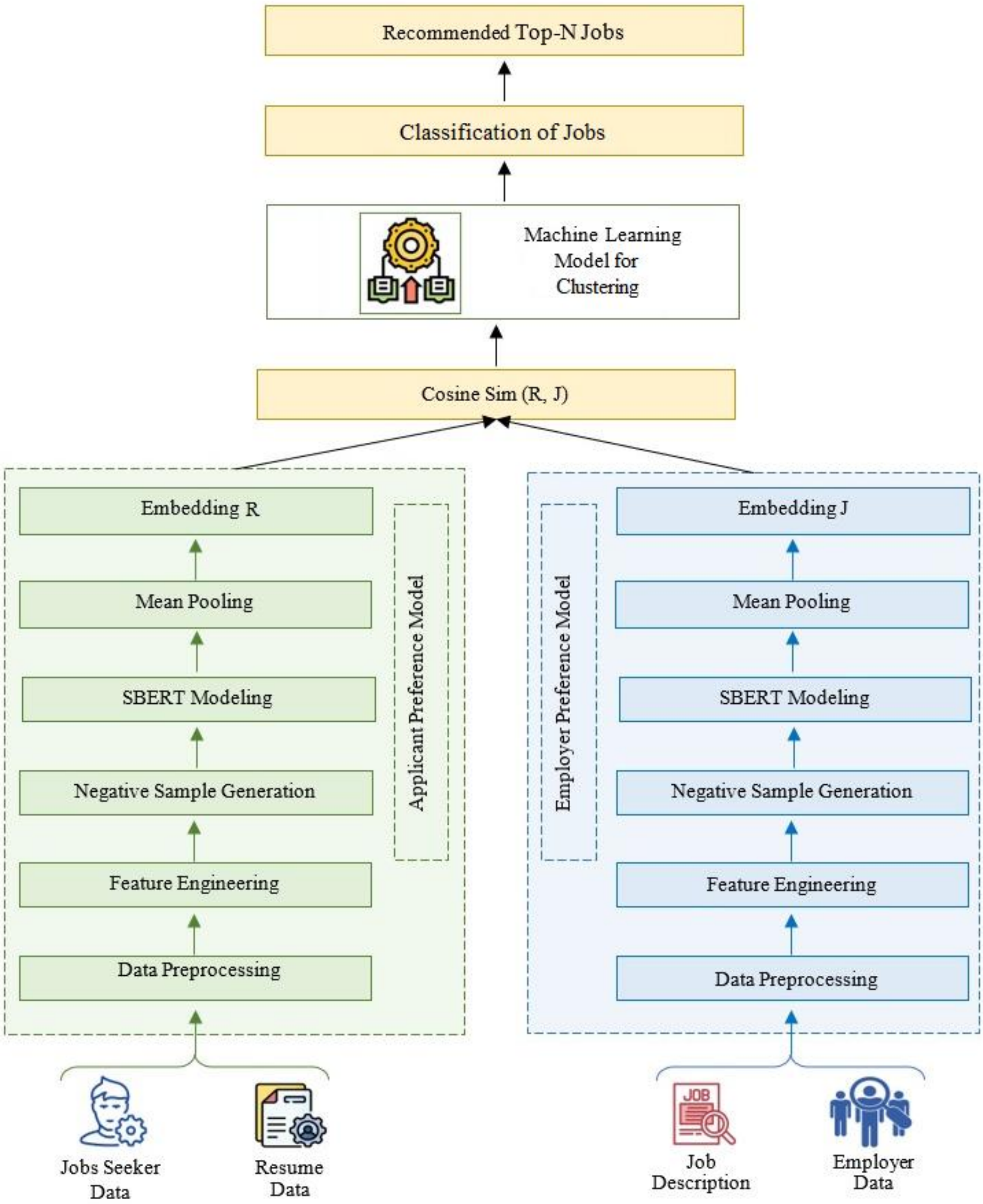

Figure 2- The proposed approach for job recommendation using SBERT and Machine Learning

A recommendation model is shown in Figure 2 that employs semantic embeddings and machine learning to rank suitable and relevant jobs towards a particular CV. This model clusters the jobs for the job seeker according to their skills sets and jobs description in advertisement produce by job recruiters.

3.1 Data Acquisition

Here two types of documents are considered for embedding, one is Resumes (R) from job seekers and another one is Job Descriptions (J) from Job Recruiter. These documents are given to BERT for embedding. The approach used for the data preparation is shown in Figure 3.

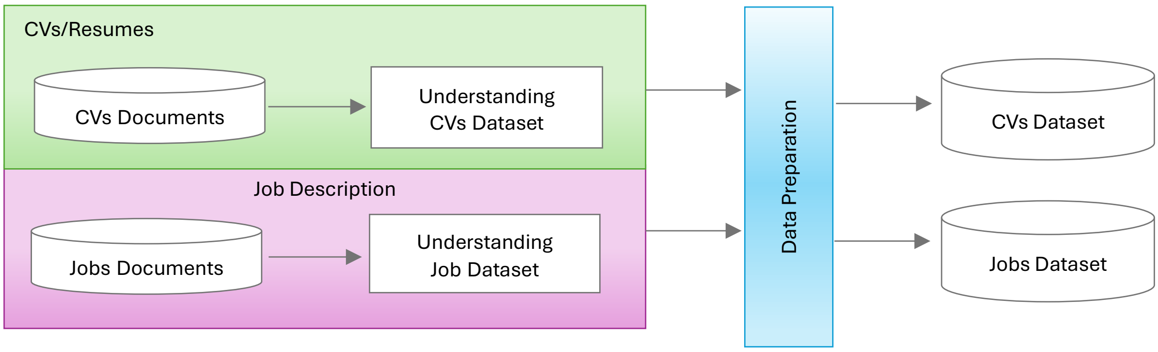

Figure 3- The proposed approach for data preparation

The following are two examples of instances of resume description and job description from each dataset. These examples are used to describe the working of the proposed approach.

A. Resume Dataset (R):

"Experienced software developer skilled in Python and machine learning. Worked on several data analysis projects and developed scalable web applications using AWS cloud services."

B. Jobs Dataset (J)

"Software Engineer needed with experience in Python, machine learning, and data analysis. Must be proficient in developing scalable web applications and working with cloud services."

3.2 Data Preprocessing

A. Text Cleaning

For text cleaning; Special symbols, unnecessary characters, and punctuation are removed from Job Description (J) documents and Resume (R) Documents.

- Job Description: "Software Engineer needed with experience in Python machine learning and data analysis Must be proficient in developing scalable web applications and working with cloud services"
- Resume: "Experienced software developer skilled in Python and machine learning Worked on several data analysis projects and developed scalable web applications using AWS cloud services"

B. Tokenization

Job Description (J) documents and Resume (R) Documents are splinted into unit.

- Job Description: ["Software", "Engineer", "needed", "with", "experience", "in", "Python", "machine", "learning", "and", "data", "analysis", "Must", "be", "proficient", "in", "developing", "scalable", "web", "applications", "and", "working", "with", "cloud", "services"]
- Resume: ["Experienced", "software", "developer", "skilled", "in", "Python", "and", "machine", "learning", "Worked", "on", "several", "data", "analysis", "projects", "and", "developed", "scalable", "web", "applications", "using", "AWS", "cloud", "services"]

C. Lowercasing

To convert into uniform documents text, Job Description (J) documents and Resume (R) Documents are converted into lowercase.

- Job Description: ["software", "engineer", "needed", "with", "experience", "in", "python", "machine", "learning", "and", "data", "analysis", "must", "be", "proficient", "in", "developing", "scalable", "web", "applications", "and", "working", "with", "cloud", "services"]
- Resume: ["experienced", "software", "developer", "skilled", "in", "python", "and", "machine", "learning", "worked", "on", "several", "data", "analysis", "projects", "and", "developed", "scalable", "web", "applications", "using", "aws", "cloud", "services"]

D. Stop words Removal

The Stop words (and, with, in, etc.) these are not significant influence are removed from Job Description (J) documents and Resume (R) Documents.

- Job Description: ["software", "engineer", "needed", "experience", "python", "machine", "learning", "data", "analysis", "must", "proficient", "developing", "scalable", "web", "applications", "working", "cloud", "services"]
- Resume: ["experienced", "software", "developer", "skilled", "python", "machine", "learning", "worked", "several", "data", "analysis", "projects", "developed", "scalable", "web", "applications", "using", "aws", "cloud", "services"]

3.3 Feature Engineering

In this research, a bi-directional analysis of data is performed as below:

A. Proposed Features for Recommending an Agency

The candidates' CV and resume data containing various (personal and technical) information of a candidate will be processed to recommend a suitable job or agency to the candidate as shown in Figure 4.

Figure 4- Recommendation of Job/Agency

The following features are used for recommending a job/agency such as workplace (city and country), nearest station, working mode, start date, salary, job position, job role and duties, requisite skills, and offer details.

B. Proposed Features for recommending a candidate

The job advertisement data containing an agency's various requirements for a job will be processed to recommend a suitable candidate for the advertised job as shown in Figure 5.

Figure 5- Recommendation of candidate

The following features are used for recommending a job/agency such as personal information, skill, qualification, experience, assessment results, work history.

3.4 Negative Sample Generation

In tasks like text analysis and job recommendations, negative sample generation can assist in creating synthetic or contrasting samples that do not basically relate to the focused group. The process of negative sample generation improves quality of NLP tasks, such as sentiment analysis, classification, and training of robust models for tasks like contrastive learning. In the proposed approach, the concept of contrastive learning is used since it supports various models like Sentence-BERT (SBERT) in distinguishing between similar and dissimilar sentence pairs. Typically used strategies for Negative Sample Generation include random sampling, semantic similarity sampling, synonym/antonym generation, etc. Since in the proposed approach, the concept of contrastive learning is used, semantic similarity sampling is considered that generates sentences that are lexically similar but contextually opposite.

3.5 SBERT Modeling

For text analytics and job recommendation, BERT (Bidirectional Encoder Representations from Transformers) model is designed that can understand the bi-directional context of words. Due to this ability of BERT, it suits such tasks of performing natural language text analysis with higher accuracy. In the proposed approach, two BERT models are used; one BERT model for Job Description (J) documents and another BERT model for Resume (R) Documents to process the given text. Each BERT processes every unit of Job Description (J) document and Resume (R) Document respectively to generate the token embedding as vector representation. Context and semantic meaning of Job Description (J) documents and Resume (R) Documents are computed by each fine-tuned BERT. Semantic meanings of each word or token in context of surrounding words are represented by vector of embedding. The following is the description of the main components of BERT and its working.

A. Transformer Architecture

The Transformer architecture is the basic building block of BERT. A self-attention mechanism is used by the transformer to process sequences of text inputs. In this implementation, BERT uses only the encoder part of the Transformer. However, the decoder part is required since it is needed for tasks with the same length of the input and output sequences.

B. Input Representation for BERT

BERT needs to input data in a particular format. For this purpose, a specific input representation is prepared for BERT. The tokens are complemented with the special tokens called CLS. These special tokens are embedded at the beginning of each text sequence (such as a sentence) for the classification purpose. In addition to that [SEP] is embedded to separate text sequences or sentences. Moreover, segment Ids are assigned to each sentence so that these embeddings can differentiate between two sentences. For example, the first sentence will get segment ID 0, and the second sentence will get segment ID 1, and so on. In addition to that positional embeddings are also employed. Such embeddings are needed because Transformers lack a built-in sense of order and these embeddings help in capturing the order of words in the input.

C. Self-Attention Mechanism

A typical BERT model needs to read the entire sequence of words at once and capture the intended relationships between words devoid of their positions. For a sequence of tokens x1, x2, ..., xn, each token in the input is projected into three vectors such as query (Q), key (K), and value (V) as shown in equation (1).

$$Q=XW_Q, \qquad K=XW_K, \qquad V=XW_V \tag{1}$$

where, the learned projections are represented using matrices like W_Q, W_K, and W_V. Afterwards, the scaled dot-product attention scores are calculated by using equation (2):

$$Attention\ (Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

where d_k is the dimension of the key vectors.

D. Multi-Head Attention

For a better performance, BERT has ability to use multiple attention heads instead of just using a single attention head. The multiple attention heads help in combinedly attending information from different subspaces of representation as shown in equation (3). Typically, the attention mechanism can be applied for the multiple times for example, 12 times in BERT-Base.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1,\text{head}_2,\ldots,\text{head}_h)W_O \tag{3}$$

where each head is computed as:

$$\text{head}_i = \text{Attention}(QW_{Qi}, KW_{Ki}, VW_{Vi}) \tag{4}$$

and W_O is a learned projection matrix.

3.6 Mean Pooling

In this step, token embedding generated by BERT, Fine tune BERT is used by applying Mean Pooling layer, the overall meaning of each Job Description (J) documents and Resume (R) Documents are computed by integrating all the tokens for averaging the embedding and convert into single fixed size vector respectively. Here, aggregation of Token Embeddings is referred to a set of fixed-size vector representations generated by Mean pooling to the token embeddings. Whereas Embedding J (Job Descriptions) are achieved by combined semantic meaning of "software engineer," "python," "machine learning," "data analysis," "scalable web applications," and "cloud services." Similarly, embedding R is combined semantic meaning of "experienced software developer," "python," "machine learning," "data analysis," "scalable web applications," and "AWS cloud services." The Mean Pooling Layer, Fixed size vector of each Job Description (J) and Resume (R) are generated by Embedding of Job Description (J) documents and Resume (R) Documents.

3.7 Generating Cosine Similarity

Cosine similarity is measured by the cosine of the angle between the vectors to indicate how much similarity exists between the job description and resume in sense of content and meaning. The function (Cosine Sim (RJ)) is used to compute the cosine similarity between the Resume (R) embedding and Job Description (J) embedding. The 1 value of cosine function of two Job Description (J) document and Resume (R) Document indicate that the documents are identical, while -1 indicate that these documents are totally different. Therefore, the relevancy can be found between Resume (R) Document of job seeker and Job Description (J) document of recruiter through cosine similarity.

- Job Description Vector (J): [0.7, 0.5, 0.6, 0.8, 0.9, 0.4]
- Resume Vector (R): [0.6, 0.6, 0.5, 0.7, 0.8, 0.5]

$$\text{Cosine Sim}(R, J) = \frac{(0.7 \cdot 0.6) + (0.5 \cdot 0.6) + (0.6 \cdot 0.5) + (0.8 \cdot 0.7) + (0.9 \cdot 0.8) + (0.4 \cdot 0.5)}{\sqrt{(0.7^2 + 0.5^2 + 0.6^2 + 0.8^2 + 0.9^2 + 0.4^2)} \cdot \sqrt{(0.6^2 + 0.6^2 + 0.5^2 + 0.7^2 + 0.8^2 + 0.5^2)}}$$

3.8 K-Means Clustering

Once the cosine similarity of a CV (R) is calculated against each job description (J), there is a need to find the top-rated jobs that match the CV. In this proposed approach, K-Means clustering algorithm is used for this purpose. K-Means helps in grouping similar job postings and recommend jobs that have the highest cosine similarity based on a candidate's CV or given preferences. To apply K-Means clustering for job recommendations, the normalization of numerical features such as salary, experience, etc. was performed to improve the accuracy of the clustering process. K-Means helps in clustering relevant jobs into groups based on similar features. Furthermore, it also helps users in identifying the nearest job cluster based on their profile.

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{5}$$

where,

J is the total within-cluster variance.

k is the number of clusters.

C_i is the set of points (cluster) assigned to cluster i.

x is a data point in cluster C_i.

μ_i is the centroid of cluster i.

$\|x - \mu_i\|^2$ is the squared Euclidean distance between point x and centroid μ_i.

3.9 Classification of Relevant Jobs

To validate the output of K-Means clustering algorithm, a rule-based classification was applied on the cosine similarity data. For classification the following threshold was defined:

(0.0, 0.249) → Highly irrelevant Jobs

(0.25, 0.349) → Medium Irrelevant Jobs

(0.35, 0.449) → Slightly Irrelevant Jobs

(0.45, 0.549) → Borderline Jobs

(0.55, 0.649 → Slightly Relevant Jobs

(0.65, 1.0) → Highly Relevant Jobs

3.10 Recommended Top N Jobs

Ranking is applied on all the jobs, all jobs are arranged according to the similarity scores jobs in accessioning order, then recommender recommend the top n jobs those most like resume are recommended to the job seeker according to their interest. Ranking is performed based on similarity scores and Top N jobs are recommended to job seekers.

4. RESULTS AND DISCUSSION

4.1 Experiment Details

The proposed deep learning based approach to solve the problem of job recommendations is discussed in this section.

A. Dataset

A set of Applicant CVs were processed against the Job Description dataset available at Kaggle. The used dataset for job descriptions title Job_Posting_Dataset has total records 9380. Figure 6 shows the cleaned and selected version of the used dataset. This job dataset has job descriptions of Jobs related to Information Technology, software development, and similar disciplines.

	A	B	C	D
1	**job_title**	**company**	**job_summary**	**job_skills**
2	C# Software Engine	E Tech Group	At E Tech Group,	['C#', '.NET', 'WPF', 'ASP.NET MVC'
3	Software Implemer	Kardex	Kardex Remstar is	['Software Implementation', 'Softwa
4	Senior Software En	Jobs for Humanity	Company	['Java', 'Python', 'SQL', 'Node.js', 'G
5	Senior Manager, Sc	Jobs for Humanity	Company	['JavaScript', 'Java', 'TypeScript', 'S(
6	Lead Software Engi	Jobs for Humanity	Company	['JavaScript', 'Java', 'AWS', 'Vue.js',
7	Sr. Software Engine	BlueCross BlueShie	Provides	['Application design', 'Analysis', 'Pr
8	Lead Software Engi	Jobs for Humanity	Company	['Java', 'Python', 'Angular', 'React', '
9	Lead Software Engi	ClickJobs.io	77 West Wacker Dr	['Java', 'Python', 'Angular', 'React', '
10	Senior Software En	FDS Avionics Corp.	Please note this	['Linux', 'U Boot', 'Low level drivers',
11	Senior Software En	ArgonFDS	Please note this	['Linux', 'UBoot', 'Low Level Drivers',
12	Senior Software En	Leidos	Description	['Software Engineering', 'Design', 'D
13	Senior Software En	ClearanceJobs	R-00123572	['Software Engineering', 'Design', 'D
14	Software Engineer	rideOS (acquired by	Who We Are	['Java', 'Algorithms', 'Simulations', '
15	Java Software Engi	Gamma Technologi	Gamma	['Java', 'Python', 'C++', 'Agile Develc
16	Software Quality Er	Life Fitness	An empty treadmill	['Software Quality Engineering', 'Tes

Figure 6 – Job Description Dataset

4.2 Results and Discussion

A. Experiments with CVs from Relevant Discipline

The first experiment was conducted by processing an IT based CV. The results in Figure 8 to Figure 11 give a broad picture of the analysis of the job recommendation system and its attempt to match the candidate to the job using similarity measures. The cosine similarity heatmap[10] shows comparison results of CVs and JDs in Figure 7. A moderate degree of overlap can be seen in the CVs and job descriptions; this means that the features can be fine-tuned, or new categories can be developed for better definition of job features. In other words, recruiters can focus on improving the choice of the candidate accordingly the position offered. In conclusion, the heatmap can aid in quickly appreciating the extent to which CVs match with the jobs and information from the heatmap can help in enhancing the recommendation system based on revelation of regions where CV-job matches are poor or average.

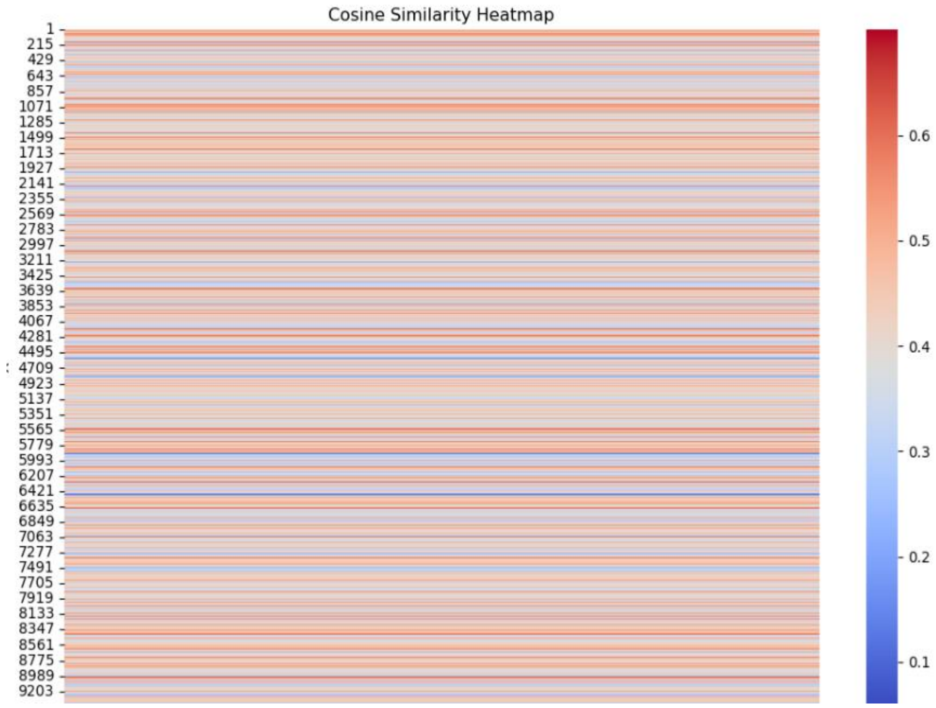

Figure 7 – Heatmap of the cosine similarities calculated for a CV 10089434 against the Dataset Job Dataset

After applying K-Means clustering, the similar index clusters scatter plot was generated that is shown in Figure 8 and it reveals the distribution of CVs with respect to job description relevance. The values indicated along the x-axis represent CV IDs, while the y-axis has a similarity index, with a scale of 0.0 to 0.7. In this plot, each data point is further marked by its cluster color, and crosses mark the positions of cluster centers in red. Based on the analysis, seven most relevant candidates belong to cluster 0 (green) with varying distances of similarity but an average of 0.5 to 0.7. The values of P1 in Clusters 1 (pink) lie in the intermediate range of 0.4–0.5 and are characterized as moderately relevant. Cluster 2 (orange) is 0.35 to 0.45 which is just relevant, while cluster 3 (blue) represents 0.2 to 0.35, which indicates least relevance. It gives understanding of where the clustering could be more fine-grained to enhance job matching opposite to the fact that the relevance is quite evenly distributed throughout all clusters. It also gives an idea of the locations where the clusters are centered in relation to the overall average suitability of the groups to the job postings.

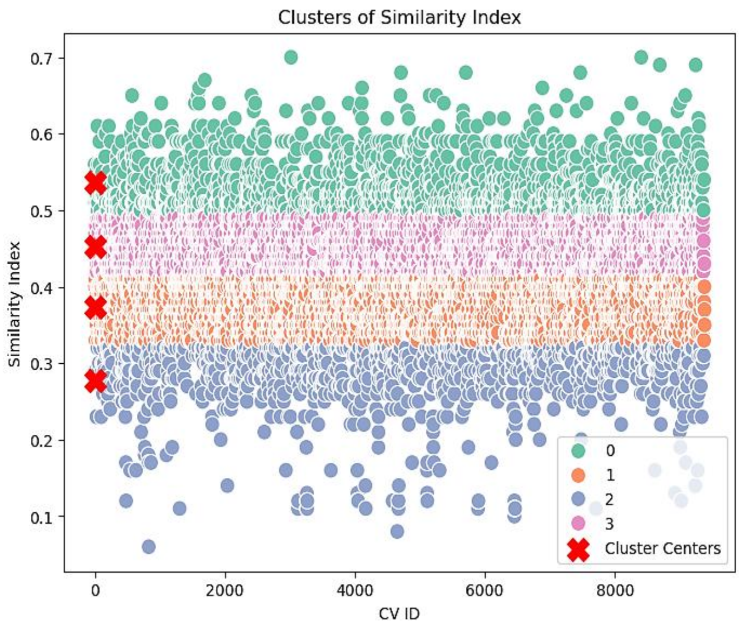

Figure 8 – Clusters generated for CV 10089434 against the Dataset Job Dataset

The box plot of similarity indices as shown in Figure 9 provides a compact description of the dispersion of similarity measures applied to CVs and job descriptions. The whiskers reach 0.1 at the bottom and 0.7 at the top of the data range, which does not include outliers. On the left side of the diagram there are several points below 0.2, it illustrates that there are businesses with low proportion of similar CVs and on the right side, there are also some points above 0.6, it illustrates that there are relevant matches. Here is depicted that most of the received CVs match job advertisements to some extent with little resemblance to extremes. Overall, the separation of CVs is relatively low, and the presence of a small IQR confirms that most of them are ranked at an average level of match. To the recruiters it means that a threshold approach could be utilised to manage candidate lists, e.g. only look at the candidates with similarity scores of above 0.5. However, the candidates with scores more than 0.7 can be highly relevant.

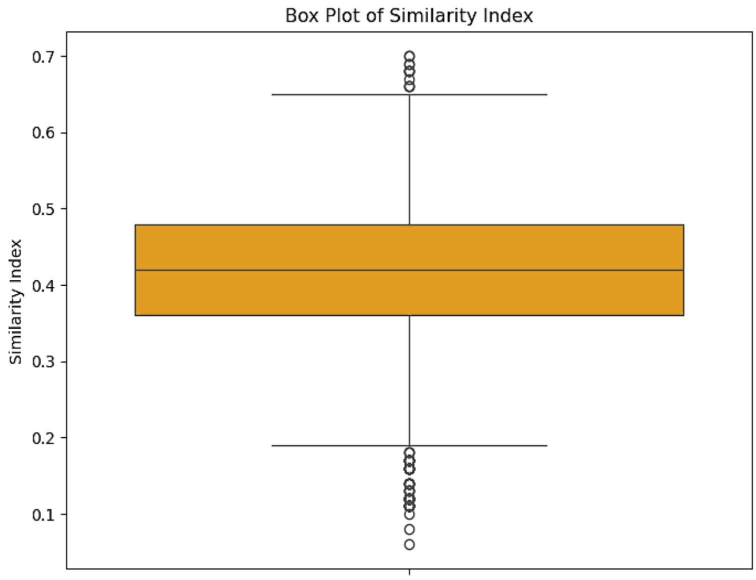

Figure 9 – Box Plot generated for CV 10089434 against the Dataset Job Dataset

A heatmap of the mean values is shown in Figure 10 that visualizes each of the clusters of CVs. The results of the cluster 0 is the highest average similarity which is 0.54 represented by deep red color that shows that this cluster has highly relevant CVs. These values reveal that Cluster 3 has a moderate relevance to the set of documents with the mean similarity of 0.45 V While the mean values of Cluster 1 and Cluster 2 are lower, 0.37, and 0.28, respectively. Based on this analysis, Mesas 2 is the least priority and can be considered as irrelevant. The variation in the similarity indices observed in the different clusters provided evidence that the clustering method was able to partition the CVs in the right groups when compared to job descriptions. It means that when recruiting people, only managers must attend Cluster 0 to identify the most suitable candidates while Cluster 3 is secondary. The distinction with color gradient helps to realize which clusters are more promising regarding the structure of recruitment increasing the effectiveness of the process. The heatmap is useful in deciding where to focus due to candidate prioritization and where in the model's adjustments could be made.

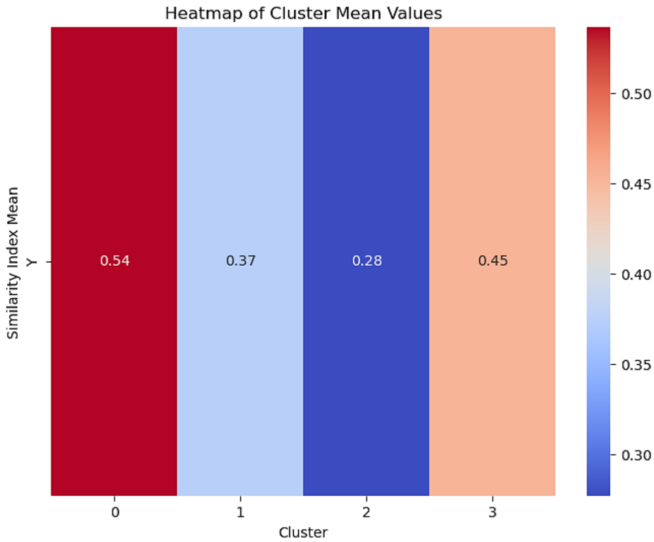

Figure 10 – Heatmap of cluster's mean values generated for CV 10089434 against the Dataset Job Dataset

A Violin plot of similarity index is shown in Figure 11 that shows the density distribution of the similarity scores for CVs with regards to the job's descriptions. This means that the width of the violin plot is defined by the values of the CVs with the most density at each of the similarity values.

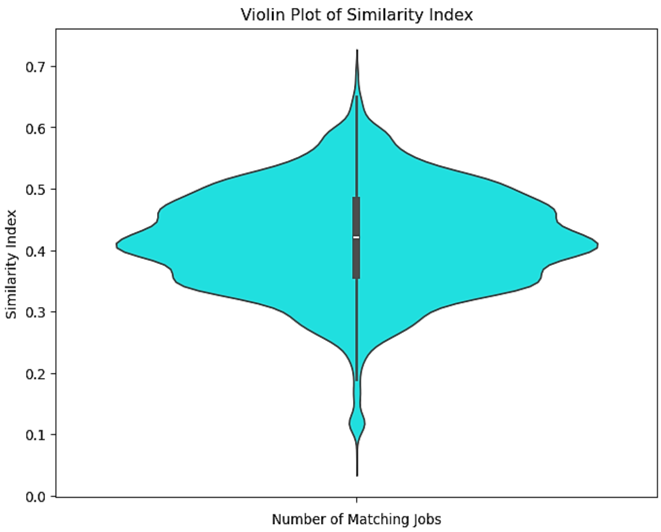

Figure 11 – Heatmap of cluster's mean values generated for CV 10089434 against the Dataset Job Dataset

For the remaining 39 sets of CVs, the median is approximately 0.4 and interquartile range implies that similarity scores range from 0.35 to 0.5 suggesting a moderate alignment of CVs to the jobs they advertised. The distribution of CVs sliding up and down the tails shows a decreased number constituting an extreme low of below 0.2 and an extreme high of above 0.6. This is epitomized in the plot which reveals that while most interfaces are moderately accurate, few are either highly matched or mismatched. This implies that," though most CVs have some match with the various jobs' requirements, extra feature sophistication may be required to enhance discrimination". Recruiters could filter CVs against other CVs and if the score is above 0.5, then consider such CVs for screening while pulling out CVs with score of less than 0.3 for better filtration of candidates.

In aggregate, the trends in the graphs indicate that there is generally moderately high candidate-job match, but there is room for enhancement in the choice of features to be used in matching, setting the match requirements, and optimizing the clustering techniques. Knowledge of these patterns will help the job recommendation system to identify better jobs for candidates and the job description to improve the efficiency of the recruitment process.

B. Experiments with CVs from Irrelevant Discipline

The second experiment was conducted by processing a CV from a non-IT (Finance) discipline. The results shown in Figure 13 to Figure 16 give a broad picture of the analysis of the job recommendation system and its attempt to match the candidate to the job using similarity measures. The Cosine similarity heatmap shows how current CVs are related on jobs descriptions where the gradient shows how similar or different they are. Here the x-axis refers to the kind of jobs or characteristics of the jobs on offer while the y-axis gives a list of the CVs or the job identification numbers. Meaning gentle shade of blue and strong shade of red imply that most of the CVs share the values between low and average with the job descriptions. The absence of peaks of large amplitude and small wavelengths implies that very few molecules show significant scores, indicating a potential for enhancements as to feature extraction or clustering techniques to achieve a clearer discrimination between highly relevant and random candidates.

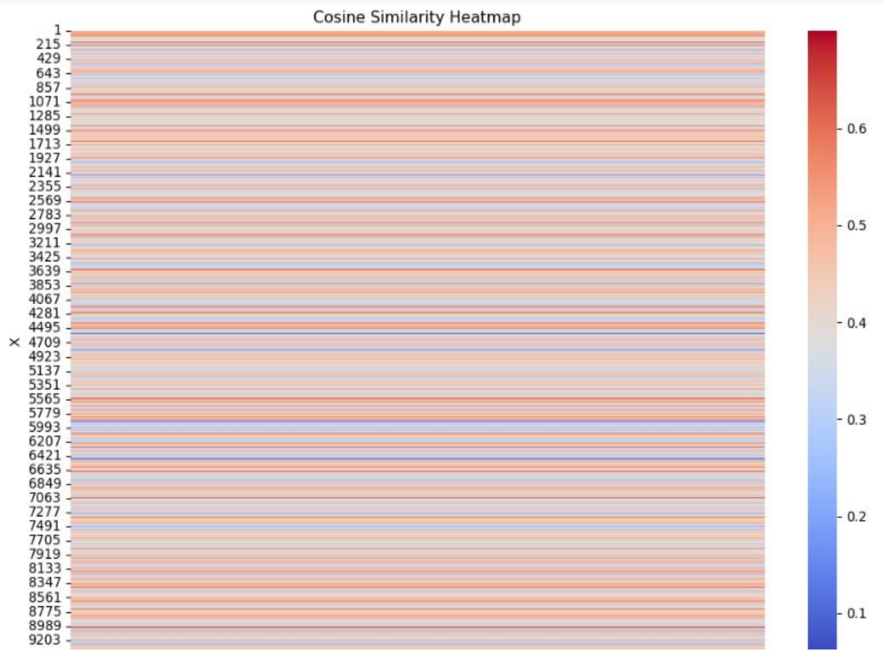

Figure 12 – Heatmap of the cosine similarities calculated for a CV 10235211 against the Dataset Job Dataset

This scatter of points indicates how it is that CVs are grouped in relation to similarity indices to job descriptions. The x-axis covers different CV IDs whilst the y-axis covers similarity indices between 0.0 and 0.6. Each of the dots is associated with the CV and clusters themselves are depicted with red crosses to indicate where the clusters are centered. In cluster three, all the CVs have close similarity measure of 0.4 to 0.5 and are the most relevant to contain an important keyword while cluster 0 contains less relevant results with similarity measure below 0.3. This clustering approach can be beneficial for the recruiters as they can complement, by concentrating on the most innovative clusters.

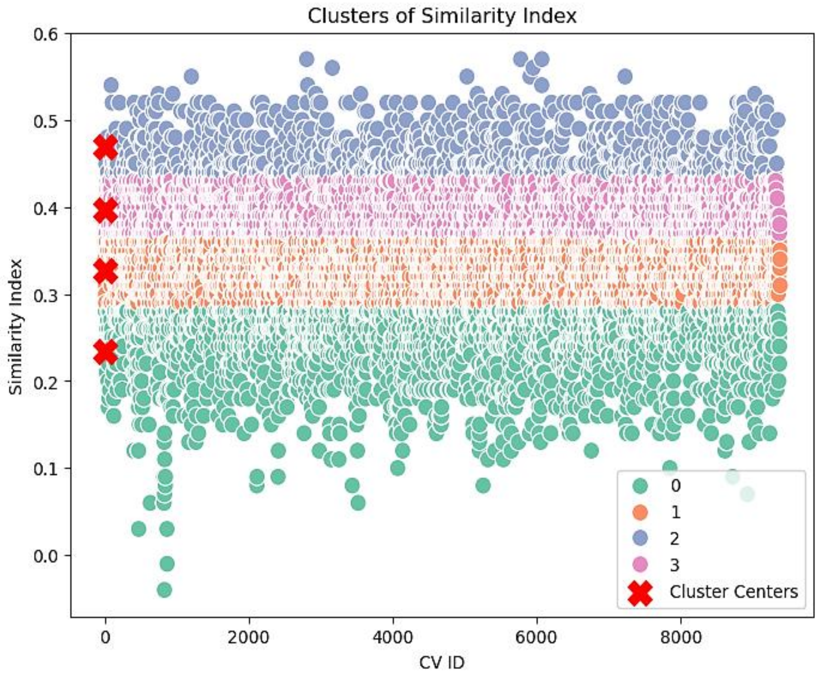

Figure 13 – Clusters of similarity index for a CV 10235211 against the Dataset Job Dataset

The box plot of similarity indices shows the following summary of the CV match with the job descriptions. It is equal to about 0.35; the IQR diagram illustrates that most of the similarity coefficients are in the region of 0.3 to 0.4. There are several extremes, the score below 0.1 and score above 0.5 which signifies that too many irrelevant and highly relevant CVs exist. This implies that the dataset has moderate Relevance in general, yet it has the capacity of filtering the desired candidate through similarity cuts for enhancement of efficiency in candidates selection.

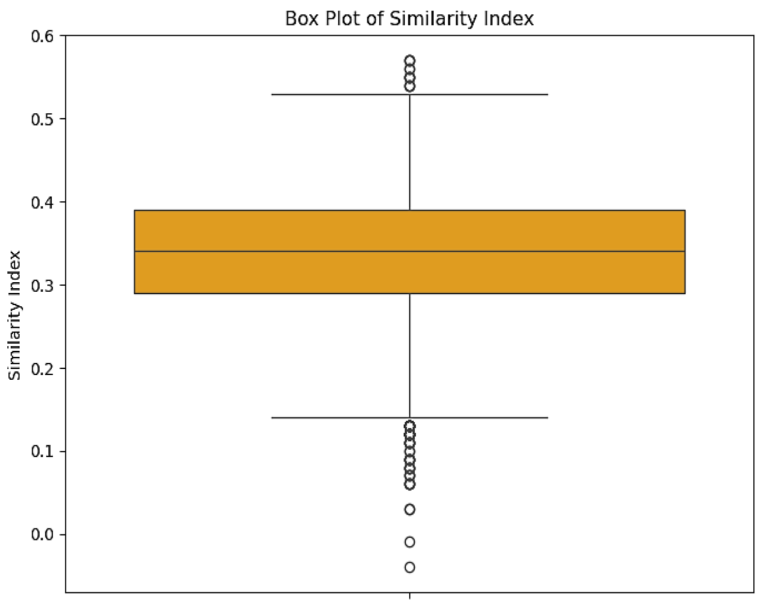

Figure 14 – Box plot of similarity index for a CV 10235211 against the Dataset Job Dataset

Clusters are marked with numbers and this heatmap shows the average similarity value from 0.23 to 0.47 only for clusters. For the first analysis, Cluster 2, represented in red in the figure, has the highest mean similarity, and can therefore be assumed to provide CVs with a higher relevance when compared to the other clusters. It shows that the CVs in Cluster 0 are the least relevant, and the mean similarity of Cluster 0 is the smallest and colored in deep blue.

This type of representation assists in quick completion of searches in a way that highlights specific clusters which may in fact include better suited candidates for a given job description to which the recruiter can then allocate his or her resources.

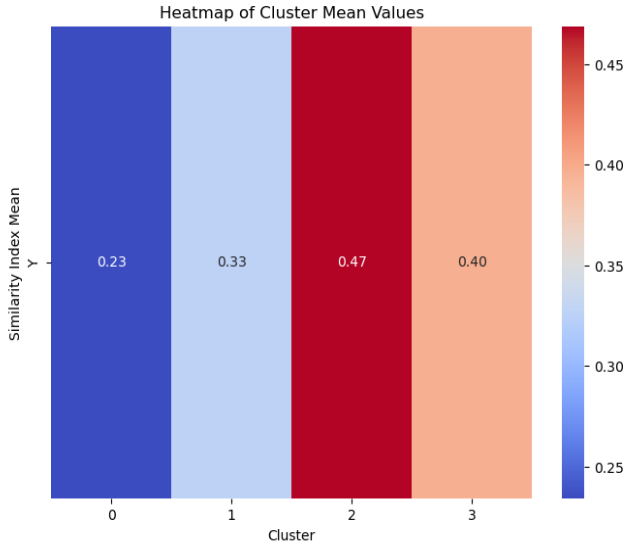

Figure 15 – Heatmap of clusters mean values for a CV 10235211 against the Dataset Job Dataset

The violin plot represents the distribution density of similarity scores in all CVs. The median is about 0.35 and most of the density is between 0.3 to 0.4, that reveals that most of the candidates have medium degree of match to the jobs. The tails of the plot go well below 0.1 and rise above 0.5 which points to the existence of vague and relevant CCR values that the algorithm seeks to eliminate from consideration. This distribution of matches is also symmetric, indicating that the data is balanced, and moderate match affiliation is the most frequent. It suggested that to simplify the recruitment process it is possible to set thresholds and consider candidates above a specific similarity level only.

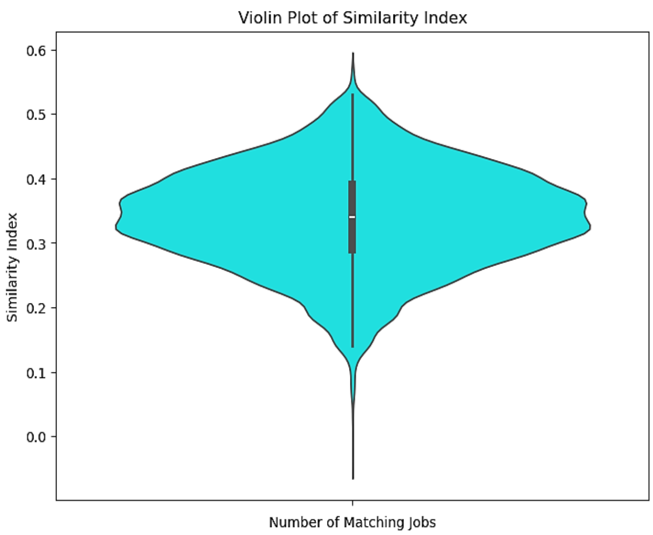

Figure 16 – Heatmap of clusters mean values for a CV 10235211 against the Dataset Job Dataset

4.3 Classification of Relevant Jobs

For classification threshold was defined given in Section 3.9 and it was used here to classify the jobs. The points plotted on the graph are actual representations of the similarity index between a set of job postings and a target CV, giving information as to how well the CV matches each of the postings. The semi-axes x are the job IDs, and the y-axis is the similarity scale from the values 0.0 to 0.7. The data points are color-coded to represent different similarity ranges: cyan for a value between 0.0 and 0.149, pink for 0.15 to 0.249 and consistently as the value increases in increments of 0.1 up to red color for the value between 0.65 and 1. The scatter plot gives a distribution of similarity scores, and a greater number of jobs clustered in the score range of 0.25 to 0.5. The red circles on top qualify as the highest relevance matching, explaining why some jobs match this target CV much better than others; cyan and pink circles being the least matching. Indeed, the absolute and overall distribution reveals distributions that are relatively evenly split into low, medium, and high match scores, which points to potential for improving upon the feature extraction and similarity functions for increasing relative high relevance.

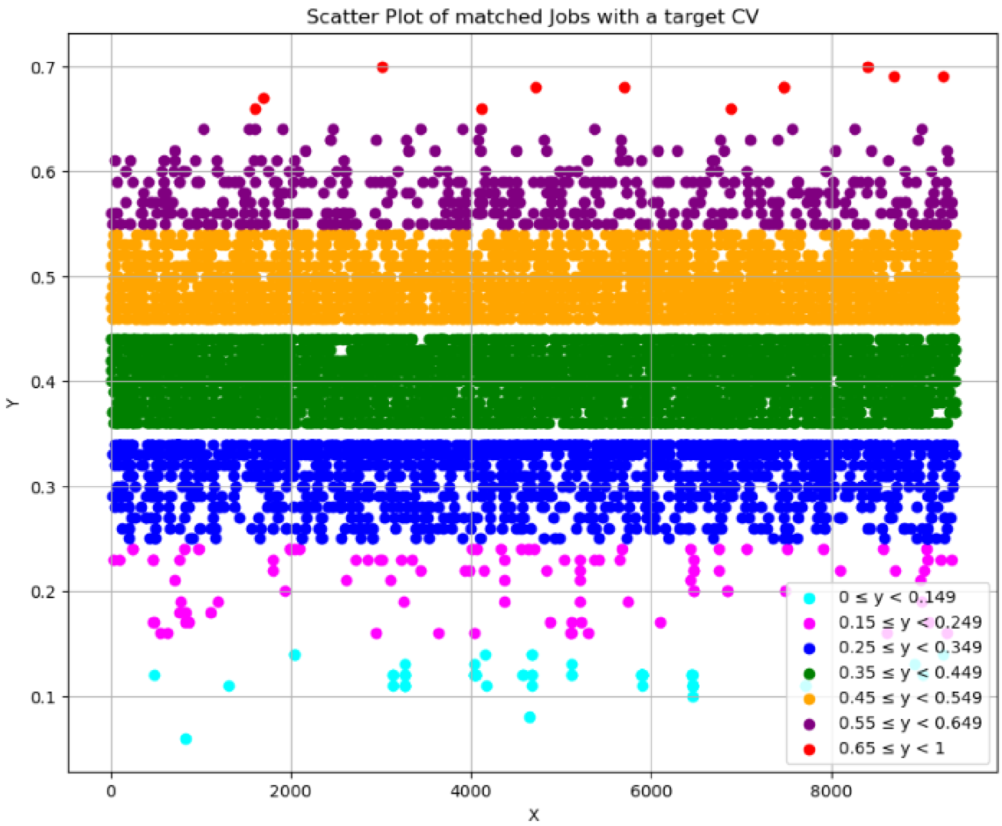

Figure 17(A) – Classification of jobs for a CV 10089434 against the Dataset Job Dataset

This scattergram presents the results of comparison between CVs and job postings, with the CV identification numbers on the horizontal axis, and the similarity scores on the vertical axis. The similarity score varies between 0.0 and 0.6 and each point is marked with color meaning certain range of scores. For the points that marked low similarity the color used is cyan and pink while for moderate to high similarity the color used are blue, green, orange and red respectively. A few of them are almost at zero, while most of them gather around the middle region from 0.25 to 0.5 which means that most CVs are aligned moderately with the job descriptions. High relevance rated matches are represented by orange and red points while poor relevance is represented by cyan points. It also enables a determination of how well CVs are distributed in relation to the job descriptions. The spread of points demonstrates that, on average, there are better CV matches and many moderate-quality CVs. Any refinement of the clustering process or the extraction of features could facilitate the making of more meaningful clusters and hence match highly suitable candidates with relevant CVs for the recruitment process.

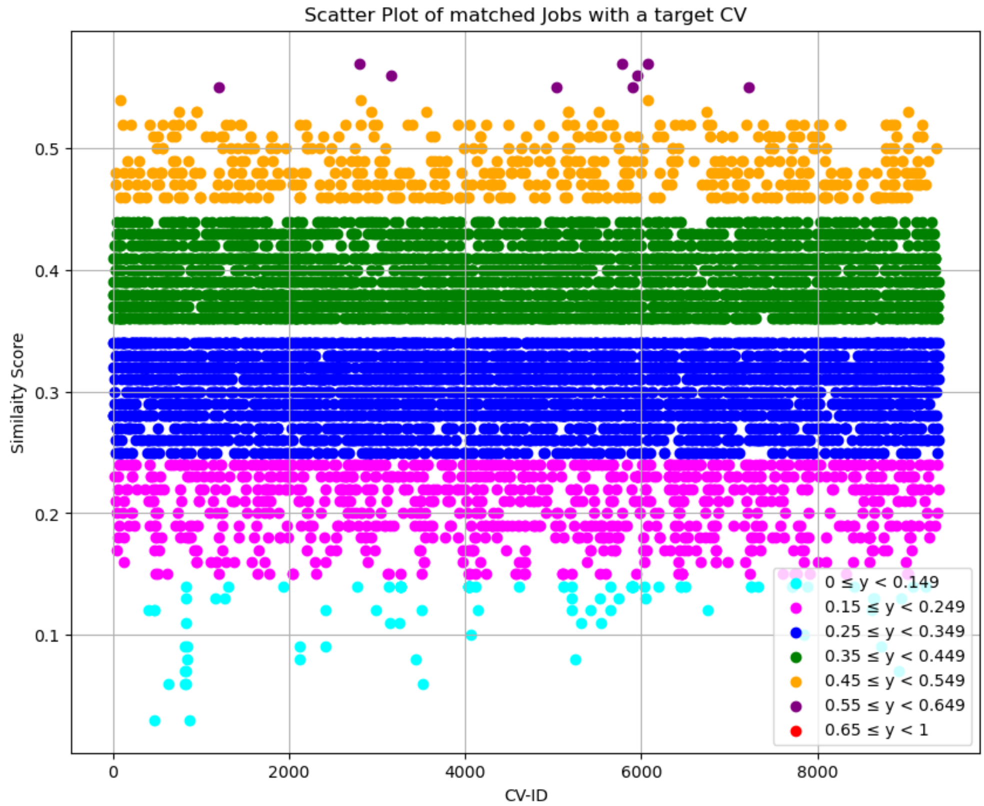

Figure 17(B) – Classification of jobs for a CV 10235211 against the Dataset Job Dataset

Figure 18 shows a Violin plot that was created to show the probability density of the similarity scores of clustered IT, Finance, and Business jobs. The horizontal axis depicts various jobs, and the vertical axis indicates similarity ranges from 0.0 to 0.7, only. The width at each violin shape indicates the distribution of how similar each category and cluster is with the other. The regions with increased density show medium level of matches, so moderate matches thus apply to most of the categories. The white dot in each violin stands for the median of the similarity scores, and the black line for the IQR.

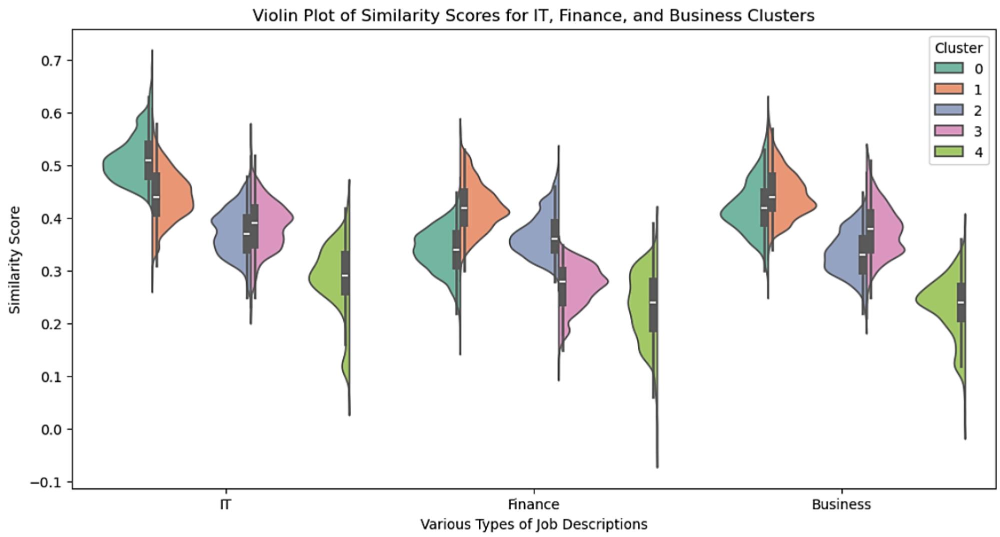

Figure 18 – Violin plot for CVs from three different disciplines against the IT Job Dataset

IT and Business are bigger as compared to other categories revealing higher variance in relevance of these topics; Finance is relatively smaller revealing that all the articles within this category are quite similar in terms of relevance to the topic. The "Violin Plot" also offers some understanding of how the similarity scores were distributed in contained types of jobs and to which extent they could be matched to CVs. It implies that for some categories, scores are lower than average, but the distribution of similarity is somewhat bimodal, with more candidates clustered around moderate similarity scores, which indicates an opportunity to improve feature harvesting for improving matches of candidates with jobs.

For further understanding of the results, Figure 19 shows a box plot that gives the statistical summary of the similarity scores for the different jobs and clusters namely IT, Finance and Business. The horizontal axis is the types of job and vertical axis is the similarity index varying between 0 and 0.7. The data boxes represent the IQR and the black line within the boxes indicates similarity scores of the medians.

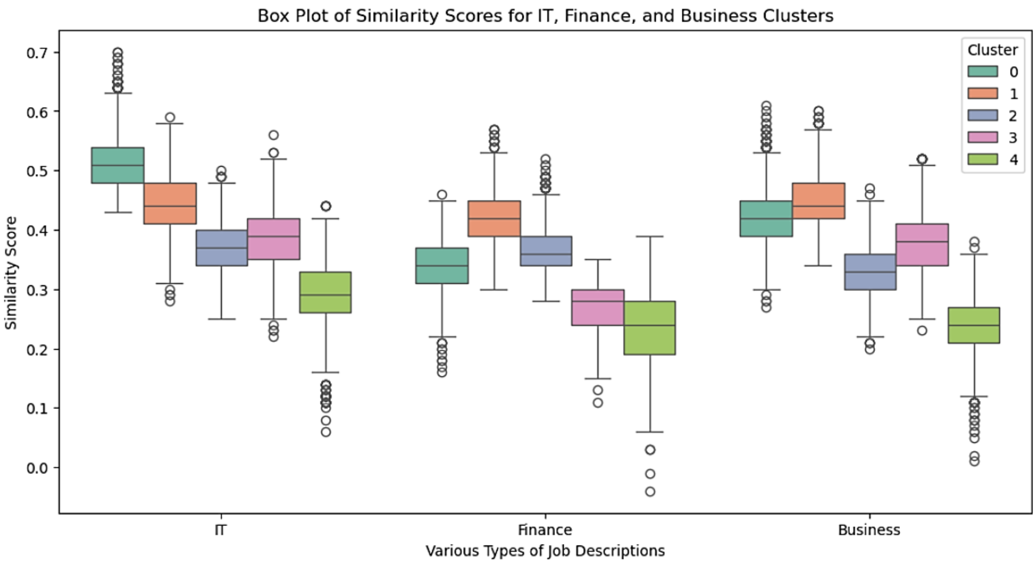

Figure 19 – Box plot for CVs from three different disciplines against the IT Job Dataset

The lines on top and bottom of bars come down to illustrate the dispersal of similarity scores in every category, and individual points of dissimilar scores are plotted outside the range of whiskers. In IT and Business subcategories, the distribution of the relevance scores is even more dispersed, evidenced by the greater IQR. In this case, the boxes are narrower than in other categories Except the Ratio Analysis box where the variation of the similarity score is wider and fluctuates a lot facing up and down and up movements contrast to other CVs in the Finance category with most of the boxes in the similar range and the fluctuation of the similarity score is relatively less dramatic from one CV to another than that of the Ratio Analysis box. Specifically, the median similarity scores IT and Business are bigger than Finance, which illustrates that CVs in the categorization could be more suitable for the job postings. The significant variability in the observed values disclosed the existence of both highly relevant and highly irrelevant CVs in the current dataset. From the box plot, it becomes easier to identify where changes may be made to enhance the standardization and, therefore, relevance of clusters to achieve better match quality

4.4 Discussion

From the offered visualizations, we get additional insights into the current performance of the given job recommendation system, which focuses on CV-job description matching across categories and categories of similarity. Every graph is different, which allows, on the one hand, to emphasize the successful aspects, and, on the other hand, to identify the aspect for development. The densities of matched jobs show how similarity scores are distributed in the CVs to job postings pairs. The majority of obtained results are in the range of 0.25 to 0.5, which shows a moderate level of relevance. The red and orange circles of points lie in the plane representing high similarity match while the cyan and pink circles of points are of poor matches. Such distribution indicates that while there are many CVs that match the content of the job postings with a certain degree of relevance, there are also many moderately relevant CVs. It is for this reason that it is possible to elevate the number of the most suitable hits modifying only the feature extraction or clustering algorithms, and, therefore, the quality of candidate-job matching.

The violin plot gives further information about the distribution density of similarity scores of different job categories: IT, Finance and Business) in different clusters. The median of the raw similarity scores from each pair of job categories are typically in the vicinity of 0.40, hence moderate extent of alignment. From the violin plots of all the categories being compared – IT and Business seem to have higher variability than Finance with relatively tighter violin plots that implies similarity scores are generally closer to each other in Finance than in IT and Business. This suggests that for categories IT and Business, there might be a desire for an increase in feature extraction to avoid potentially high variability of scores on the top hits and better target them.

5. CONCLUSION AND FUTURE WORK

A novel and intelligent job recommendation system is proposed in this paper. The designed job recommendation system is based on semantic embeddings and machine learning for clustering the relevant jobs. The top-N jobs are achieved by cluster with the highest cosine similarity score. The results of the experiments manifest that general distributions for various job types help in identifying the jobs relevant to a targeted CV or resume. In the achieved results; it is also identified that the median, the interquartile range (IQR), and outliers are depicted for every kind of job and for clusters, indicating variation in alignment. The percentages of matches in IT and Business categories have more IQRs hence implying more variability than the Finance category. Many outliers indicate both important and unimportant CVs so, there is a tendency to refine the clustering or use other features for the elimination of irrelevant candidates. In general, these visualizations demonstrate that most of the CVs are rather like job ads from the medium range with few highly similar positions. Optimization of feature extraction, clustering and similarity could improve match quality making it easier for the recruiter to prioritize candidates of high relevance. This would improve the efficiency of the job recommendation system as well as the general quality of job-candidate matching.

To improve the further quality of this job recommendation system, the algorithm can be further improved to increase the similarity index of the job descriptions. Also, the system can be improved to suggest the top n number of jobs explaining for each job that this job is n% similar to the uploaded CV and which area the candidate should further focus on to have better chances of recruitment.

REFERENCES

[1] Kino, Y., Kuroki, H., Machida, T., Furuya, N., & Takano, K. (2017). Text analysis for job matching quality improvement. Procedia computer science, 112, 1523-1530.

[2] Liu, R., Rong, W., Ouyang, Y., & Xiong, Z. (2017). A hierarchical similarity-based job recommendation service framework for university students. Frontiers of Computer Science, 11, 912-922.

[3] Yang, S., Korayem, M., AlJadda, K., Grainger, T., & Natarajan, S. (2017). Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach. Knowledge-Based Systems, 136, 37-45.

[4] Sridevi, G. M., & Suganthi, S. K. (2022). AI-based suitability measurement and prediction between job description and job seeker profiles. International Journal of Information Management Data Insights, 2(2), 100109.

[5] Chou, Y. C., & Yu, H. Y. (2020, August). Based on the application of AI technology in resume analysis and job recommendation. In 2020 IEEE International Conference on Computational Electromagnetics (ICCEM) (pp. 291-296). IEEE.

[6] Guo, S., Alamudun, F., & Hammond, T. (2016). RésuMatcher: A personalized résumé-job matching system. Expert Systems with Applications, 60, 169-182.

[7] Keim, T. (2007, January). Extending the applicability of recommender systems: A multilayer framework for matching human resources. In 2007 40th Annual Hawaii International Conference on System Sciences (HICSS'07) (pp. 169-169). IEEE.

[8] Mhamdi, D., Moulouki, R., El Ghoumari, M. Y., Azzouazi, M., & Moussaid, L. (2020). Job recommendation based on job profile clustering and job seeker behavior. Procedia Computer Science, 175, 695-699.

[9] Yadalam, T. V., Gowda, V. M., Kumar, V. S., Girish, D., & Namratha, M. (2020, June). Career recommendation systems using content-based filtering. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)* (pp. 660-665). IEEE.

[10] Rashid, A. H. A., Mohamad, M., Masrom, S., & Selamat, A. (2022, September). Student Career Recommendation System Using Content-Based Filtering Method. In *2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS)* (pp. 60-65). IEEE.

[11] Kara, A., Daniş, F. S., Orman, G. K., Turhan, S. N., & Özlü, Ö. A. (2022, September). Job Recommendation Based on Extracted Skill Embeddings. In *Proceedings of SAI Intelligent Systems Conference* (pp. 497-507). Cham: Springer International Publishing.

[12] Wu, L., Qiu, Z., Zheng, Z., Zhu, H., & Chen, E. (2024, March). Exploring large language model for graph data understanding in online job recommendations. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, No. 8, pp. 9178-9186).

[13] Shi, X., Wei, Q., & Chen, G. (2024). A bilateral heterogeneous graph model for interpretable job recommendation considering both reciprocity and competition. *Frontiers of Engineering Management*, *11*(1), 128-142.

[14] Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., & Seveso, A. (2021). Skills2Job: A recommender system that encodes job offer embeddings on graph databases. *Applied Soft Computing*, *101*, 107049.

[15] Liu, R., Ouyang, Y., Rong, W., Song, X., Tang, C., & Xiong, Z. (2016). Rating prediction based job recommendation service for college students. In *Computational Science and Its Applications–ICCSA 2016: 16th International Conference, Beijing, China, July 4-7, 2016, Proceedings, Part V 16* (pp. 453-467). Springer International Publishing.

[16] Yang, S., Korayem, M., AlJadda, K., Grainger, T., & Natarajan, S. (2017). Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach. *Knowledge-Based Systems*, *136*, 37-45.

[17] Reusens, M., Lemahieu, W., Baesens, B., & Sels, L. (2017). A note on explicit versus implicit information for job recommendation. *Decision Support Systems*, *98*, 26-35.

[18] Almalis, N. D., Tsihrintzis, G. A., & Papaioannou, I. (2020). Handling the crowd avoidance problem in job recommendation systems integrating FoDRA. *International Journal of Computational Intelligence Studies*, *9*(1-2), 128-145.

[19] Wang, Y., Allouache, Y., & Joubert, C. (2021, December). A Staffing Recommender System based on Domain-Specific Knowledge Graph. In *2021 Eighth International Conference on Social Network Analysis, Management and Security (SNAMS)* (pp. 1-6). IEEE.

[20] Heggo, I. A., & Abdelbaki, N. (2021). Data-driven information filtering framework for dynamically hybrid job recommendation. In *Intelligent Systems in Big Data, Semantic Web and Machine Learning* (pp. 23-49). Cham: Springer International Publishing.

[21] Rivas, A., Chamoso, P., González-Briones, A., Casado-Vara, R., & Corchado, J. M. (2019). Hybrid job offer recommender system in a social network. *Expert Systems*, *36*(4), e12416.

[22] Yao, J., Xu, Y., & Gao, J. (2023). A Study of Reciprocal Job Recommendation for College Graduates Integrating Semantic Keyword Matching and Social Networking. *Applied Sciences*, *13*(22), 12305.

[23] Reusens, M., Lemahieu, W., Baesens, B., & Sels, L. (2018). Evaluating recommendation and search in the labor market. *Knowledge-Based Systems*, *152*, 62-69.

[24] Chen, J., Yuan, B., Jin, C., Xie, W., Wang, J., & Zhu, R. (2022, August). Design and Implementation of Employee Recommendation System Based on Neural Graph Collaborative Filtering. In *International Conference on Image, Vision and Intelligent Systems* (pp. 784-792). Singapore: Springer Nature Singapore.

[25] Prince, D., Madhan, K., Vishwa, K., & Yamunathangam, D. (2023, June). Job and Course Recommendation System using Collaborative Filtering and Naive Bayes algorithms. In *2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)* (pp. 1-4). IEEE.

[26] Zhou, Q., Liao, F., Chen, C., & Ge, L. (2019). Job recommendation algorithm for graduates based on personalized preference. *CCF Transactions on Pervasive Computing and Interaction*, *1*, 260-274.