CV ANALYSIS AND OPTIMIZING THE RECRUITMENT PROCESS IN THE IT INDUSTRY USING MACHINE LEARNING TECHNIQUES

Final (Draft) Report

Maldeniya M.M.D

(IT20203726)

Bachelor of Science (Hons) Degree in Information Technology

Specializing in Data Science

Department of Information Technology

Faculty of Computing

Sri Lanka Institute of Information Technology

Sri Lanka

September 2023

CV ANALYSIS AND OPTIMIZING THE RECRUITMENT PROCESS IN THE IT INDUSTRY USING MACHINE LEARNING TECHNIQUES

Maldeniya M.M.D

(IT202003726)

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science (Hons) Degree in Information Technology Specializing in Data Science

Department of Information Technology

Faculty of Computing

Sri Lanka Institute of Information Technology

Sri Lanka

September 2023

DECLARATION

I declare that this is my work, and this dissertation does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or Institute of higher learning, and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as article or books).

Name	Student ID	Signature
Maldeniya M.M.D.	IT20203726	

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

…………………………   ………………………

Signature of the Supervisor   Date

(Dr. Anuradha Karunasena)

ABSTRACT

Identifying the right individuals is crucial for an organization's success and growth. However, traditional recruitment methods involving manual procedures, such as screening CVs, evaluating academic qualifications, and assessing technical and professional skills, are not only time-consuming but also ineffective. To meet the evolving demands of employers, it is imperative to adopt an efficient and reliable approach to assess the skills and abilities of candidates. We designed our proposed solution to optimize the hiring process in the IT industry by implementing a system that can accurately and effectively find the candidates who are best suited for a given job role. In this research component, the primary objective is to rank candidates by utilizing a job description generated through a structured format. By using a proper job description, the candidate's skills can be easily identified, making the ranking process more efficient. Additionally, to provide recruiters with a better understanding of the candidate, graphical representation will be provided. Overall, this system leverages advanced machine learning algorithms and natural language processing to analyze candidate data and match it with the job description, resulting in a streamlined and objective evaluation process.

Keywords – CV, Machine Learning, Ranking, Natural Language Processing, job description, Recruiting

ACKNOWLEDGEMENT

I would like to offer my sincere gratitude to everyone who supported me in making my final-year research project a success. Firstly, I would like to extend my sincere gratitude to my project supervisor Dr.Anuradha Karunasena, and co-supervisor Dr. Lakmini Abeywardhane, for their guidance and the invaluable knowledge shared throughout the year to make the research successful. The success of this research would not have been possible without their support and constant guidance.

I would also like to express my gratitude to the Sri Lanka Institute of Information Technology and the CDAP team for providing us with the guidance and the expertise to finish the project successfully. I also sincerely thank all the project group members, who worked with me and supported me to complete this research successfully.

Maldeniya M.M.D
Faculty of Computing,
Sri Lanka Institute of Information Technology.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CV	Curriculum vitae
IT	Information Technology
JD	Job Description
ICT	Information Communication Technology
ICTA	Information and Communication Technology Agency of Sri Lanka
NLP	Natural Language Processing
BERT	Bidirectional Encoder Representations from Transformers
MLM	Machine Learning Model
HR	Human Resources
SDLC	Software Development Life Cycle
IDE	Integrated Development Environment
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
DB	Data Base
API	Application Programming Interface
GPU	Graphics Processing Units
TPU	Tensor Processing Units

1 INTRODUCTION

1.1 CV Analysis and Optimizing the Recruitment Process

Selecting the right candidates plays a pivotal role in an organization's journey towards success and expansion. However, conventional hiring methods often prove to be sluggish and ineffective, relying heavily on manual tasks like sifting through resumes and evaluating skills. To better cater to the needs of employers, there's an imperative need for a more efficient and precise approach to assessing the qualifications and capabilities of candidates. Unfortunately, there is currently no universally accepted standard for evaluating CVs, which is of paramount importance in choosing the ideal candidate for a specific job position. The research is focused on the IT sector, with the primary goal of streamlining the hiring process. This is achieved through the integration of cutting-edge technologies such as machine learning, data extraction, and natural language processing. By meticulously examining a candidate's technical expertise, professional aptitude, and even personality traits, hiring managers are empowered to make well-informed decisions. This ensures the selection of the most suitable candidates for available roles. This approach represents a more efficient and precise means of evaluating candidates, with a particular emphasis on CV analysis, thereby elevating the quality of hires and ultimately contributing to an organization's enduring success in the realm of recruitment.

1.1.1 Curriculum Vitae

A CV or curriculum vitae is a document that summarizes a person's educational and professional history, skills, accomplishments, and other relevant information. It is commonly used in the job application process to demonstrate a candidate's qualifications and suitability for a particular job. Customizing a CV to the job or industry is crucial. that candidates are applying for, emphasizing relevant experience and skills.

1.1.2 Job Description

A job description is a written statement that specifies the job responsibilities, duties, necessary skills, qualifications, and any other relevant requirements for an employee working in a specific position or role within an organization. It provides a clear and concise summary of the tasks, expectations, and performance standards that are associated with a job.

1.1.3 Research Area

Numerous studies have been conducted to assess the skill proficiency of candidates from various perspectives. Researchers have ranked the candidate pool to get the best fit, with a focus on analyzing and evaluating their resumes. Additionally, this study aims to determine if this approach can assist recruiters in identifying top talent more quickly and effectively.

1.1.4 Component Overview

The main objective of this component is to review candidates' resumes and identify the most qualified individual for a specific job vacancy. The emphasis is on using a well-defined job description to rank the CVs and obtain optimal outcomes. Through this component, a comparison between the job description and the candidate's skills will be presented using a summary representation, making it easy to identify even for those with limited computer literacy.

1.2 Background Literature

1.2.1 Background Study

The ICT sector in Sri Lanka is thriving with innovative technological advancements, showcasing the natural intelligence of its people. With the second-highest revenue, Sri Lanka's IT industry significantly contributes to the nation's exports, connecting more than 500 industries with information technology. This responsible and sustainable industry highly values labor laws and environmental regulations. Due to a skilled workforce and cost-effective operations, the IT sector has become incredibly profitable. [1]According to a report by the ICTA, Sri Lanka's IT industry has been growing at a rate of 16% per year. By 2022 [2], it is projected to create more than 200,000 jobs both directly and indirectly. The nation's robust talent pool of IT experts, who have received top-notch education and training, make Sri Lanka competitive in the international IT business. Despite the high demand for IT employment, obstacles exist in selecting the best applicants to fill these positions.

The Workforce survey findings present a compelling narrative of substantial growth within the job market. [3]Over the course of four years, from 2014 to 2018, the workforce in the surveyed region surged impressively, escalating from 82,854 individuals to 124,873. This remarkable upswing signifies a substantial growth rate of 50.7%, indicative of a vibrant and expanding job landscape. What's even more promising is that this growth trend appears poised to persist, with projections indicating a further leap to 146,089 in 2019, painting a picture of continued opportunity and employment prospects. Notably, the Information and Communication Technology (ICT) sector stands out as a pivotal driver behind this workforce expansion. It occupies a central role in this narrative, with a significant 92% of employees actively engaged in emerging technologies, underscoring the sector's rapid and transformative development. This is a testament to the sector's adaptability and ability to embrace cutting-edge innovations, further solidifying its importance in the local economy.

Moreover, the survey sheds light on the educational profile of the ICT workforce, revealing that an impressive 64.2% hold bachelor's degrees. This statistic signals a positive and encouraging trend in terms of the quality and educational qualifications of individuals working in the ICT sector. It underscores the sector's commitment to fostering a highly educated and skilled workforce, which is essential for sustaining growth and competitiveness in the ever-evolving world of technology and business.

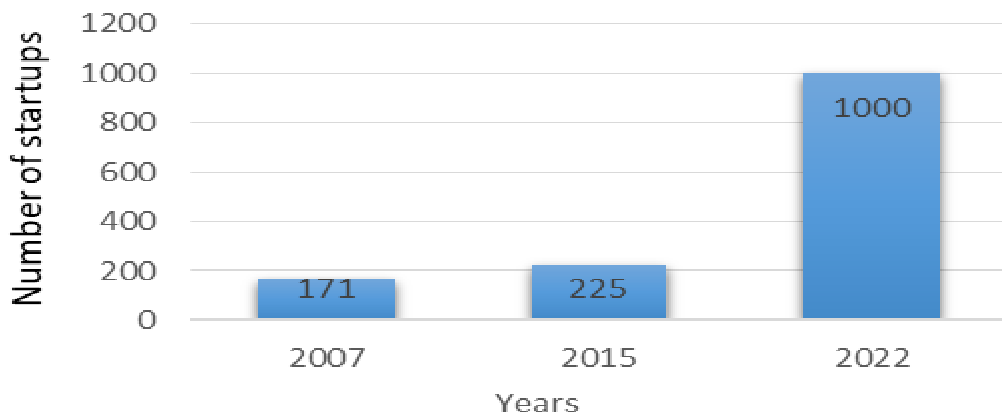


Figure 1 Growth of ICT Industry

According [4]to data provided by the Sri Lankan Export Development Board, the ICT services sector has emerged as the fourth-largest revenue contributor to Sri Lanka's economy. At present, this sector is a dynamic landscape, encompassing over 600 companies that operate across a wide spectrum of industries, including Communication, Apparel and Textiles, Banking, Financial Services, Healthcare, Manufacturing, Media, Retailing, Transportation, Travel and Leisure, among others. This diversity highlights the far-reaching impact of ICT across various domains. Considering the rapid expansion and evolution of the ICT industry, there is an escalating demand for exceptionally skilled candidates. These candidates are not only expected to possess technical prowess but also domain-specific expertise to effectively align technology with the unique objectives and challenges faced by organizations within their respective sectors. As such, candidates aspiring to excel in the Sri Lankan ICT landscape should bring a combination of technical proficiency, adaptability, and a deep understanding of industry-specific nuances to the table, enabling them to drive innovation and success in this dynamic and ever-expanding field.

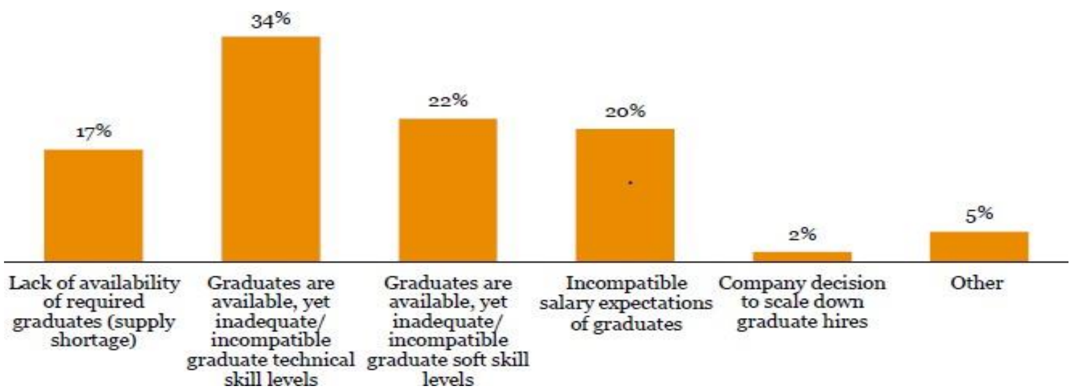


Figure 2 Skill Analysis of Graduates in Sri Lanka

In today's highly competitive global market, the success of companies hinges on their ability to make astute decisions when it comes to selecting the right candidates for IT positions. At the heart of the hiring process lies the primary objective of identifying the most suitable and qualified individuals to fill critical roles within the organization. This task is predicated on the candidates' capacity to effectively meet the specific job requirements, making it essential to conduct a comprehensive assessment of various facets of the applicants.

Ensuring a good match between an employee and a job is crucial, as it directly impacts the quality and quantity of the employee's work, as well as the overall performance and productivity of the organization. Therefore, a meticulous selection process is necessary to identify the best fit candidate for the

Finding a job in today's competitive market can be challenging, particularly if one lacks the necessary skills or presents a poorly written CV. Although there may be several reasons for this situation, individuals can take steps to enhance their chances of success and progress in their careers. As per Figure 3, [1] the absence of necessary qualifications or experience for the desired position can be a reason for unemployment due to a skills gap.

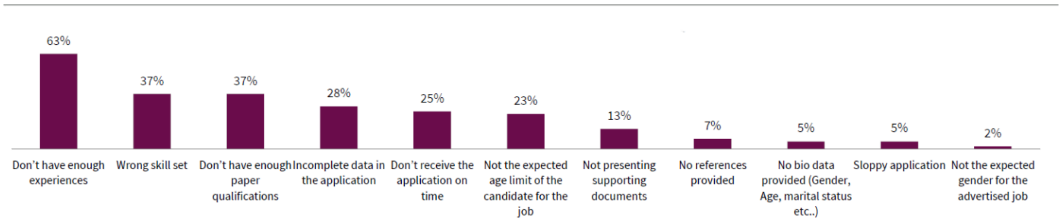


Figure 3 CV Rejection criteria analysis

A Curriculum Vitae (CV), commonly referred to as a resume, stands as an indispensable tool for individuals actively engaged in their pursuit of employment opportunities. This meticulously crafted document serves as a comprehensive summary, systematically encapsulating an individual's qualifications, skills, and professional journey. Analogously, it functions as a snapshot, providing a succinct overview of one's career trajectory. The paramount significance of the CV emanates from its role as the primary source of information that prospective employers review during the initial stages of candidate assessment.

In the intricate process of talent acquisition and recruitment, the CV assumes a central and pivotal role. Acting as a discerning filter, it aids employers in their endeavor to ascertain the suitability of a candidate for a specific role within their organization. Essentially, it represents the inaugural impression a job applicant imparts upon a prospective employer. To heighten the prospects of being considered favorably for a job opportunity, it is imperative that the content within the candidate's CV closely aligns with the prerequisites articulated within the job description. A harmonious confluence between the CV and the job's stipulated requirements serves as an indicator of the candidate's aptitude for the role, substantially augmenting the likelihood of extending an invitation for an interview.

In summation, the CV serves as the gateway to capturing the attention of discerning employers and embarking upon the inaugural steps toward securing gainful employment. Its role is emblematic of the critical interface between a candidate's aspirations and the discerning standards of modern employment practices

As a result, this approach aims to develop an automated system that can accurately rank candidate resumes according to the job description, [2] while providing skill analysis based on the resume.

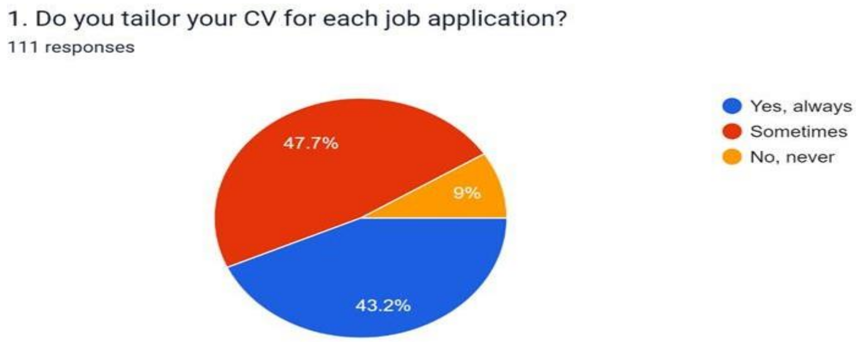


Figure 4 Survey results related to cv tailoring Job description.

1.2.2 Literature Review

The evolution of recruitment methods from traditional paper applications to digital resumes has been driven by advances in technology and changes in the job market landscape. This transition has brought about significant benefits and challenges for both job seekers and employers. As the global population continues to grow rapidly, the

competition for job vacancies has intensified, leading to a surge in the number of applicants and, consequently, an overwhelming influx of emails and digital applications for open positions. To effectively address this challenge, several research studies and innovative practices have been conducted to streamline and simplify the recruitment process.

The process [5](Research A) involves extracting pertinent information from a resume and organizing it into segments based on its values. This solution was designed specifically for unstructured CVs, as resumes can vary greatly in their format and structure. NER was used to easy the transformation from unstructured to structured resumes. A machine learning-based applicant ranking model was fine-tuned and enhanced. This involved training the algorithm on a substantial dataset to further refine its ability to construct an effective ranking algorithm. The goal was to enable the model to accurately predict the recruiter's assessment when provided with a candidate's resume.

Rasika Ransing et al. [6](Research B)utilized machine learning techniques to address the needs of recruiters. Their study focused on streamlining the resume screening process by aligning it with recruiters' specified criteria. The research dataset, presented in CSV format, contained resumes categorized with labels corresponding to various roles within the IT industry. To effectively assess candidate suitability, the researchers employed a stacked classification approach, allowing them to systematically rank resumes based on their fitness for different IT roles. This study yielded valuable insights into the significance of specific job roles within the IT industry, offering recruiters a more efficient and informed approach to talent acquisition.

In research [7](Research C) shows an automated process is employed to extract critical segments from resumes, including educational and professional experiences, along with various employment details. This transformation involves the conversion of unstructured resumes into semi-structured documents, and it is achieved through the utilization of Natural Language Processing (NLP) techniques in combination with rule-based regular expressions. Here, the system utilizes a filtration module to eliminate irrelevant terms that don't aid in matching. Additionally, a third module categorizes resumes and job posts into occupational categories using a set of skills extracted from both sources.

The paper referenced as [8] (Research D) introduces a recruitment search engine that enhances the matching process between candidate profiles and job descriptions through the strategic use of keywords. This search engine leverages advanced techniques, including BERT and Pre-trained MLM (Masked Language Model), to identify pertinent skills. Furthermore, it incorporates a unique approach by integrating CMAP (Conceptual Mapping) and competency keywords into a knowledge graph. This innovative system possesses the ability to provide valuable recommendations, particularly in the realm of topics related to neighborhoods and their associated subjects, thus significantly improving the precision and relevance of job candidate recommendations.

Automated Resume Classification System Using Ensemble Learning

In research [9] ,An ensemble model has been developed to synergize a 1D Convolutional Neural Network (CNN) and a Bi-directional Gated Recurrent Unit (GRU) for the analysis of textual data. This model comprises two separate channels, each dedicated to one of these neural network structures. The central objective of this ensemble model is to analyze text messages and associate them with specific real-world outcomes.

This research [10] “Ranking résumés automatically using only résumés” introduces an innovative approach to modern recruitment challenges by combining two key elements: conceptual-based resume classification and candidate ranking based on job postings. A substantial dataset of 2000 resumes and 10,000 job postings was meticulously curated, distinguishing it from conventional keyword-based resume analysis. Instead of relying on keywords, this approach employs conceptual classification, focusing on the underlying concepts within resumes to assess their suitability for specific job roles. Simultaneously, a ranking system considers both candidate qualifications and job posting nuances, providing a more comprehensive assessment. The results showed significantly improved precision in matching candidates with job offers, highlighting the potential of this hybrid model to enhance recruitment efficiency and effectiveness. This research is a substantial contribution to the field, offering a promising solution to connect job seekers and employers while improving hiring practices.

In this research, [11] ,the approach used relies on Natural Language Processing (NLP) techniques, specifically Word2Vec, a tool that turns words into numerical representations, capturing their meanings. Words close to each other in this numeric space are seen as having similar meanings. This method is valuable for accurately sorting resumes based on the skills they list, making it easier to find the best-fitting resumes with great precision. This research showcases the potential of NLP and word embeddings, offering a glimpse into how language analysis can be more effective in practical applications.

1.3 Research Gap

According to existing research studies, some issues were found when comparing them.

- The absence of a well-defined job description structure makes it challenging to accurately rank resumes.

The effectiveness of previous studies that ranked CVs using job descriptions extracted from various sources could be improved by utilizing a well-defined structure to generate job descriptions. Taking a structured approach to creating job descriptions is likely to yield more accurate and relevant results. This approach ensures that all relevant information is included, which can enhance the process of ranking CVs and lead to better outcomes.

- There is difficulty achieving high accuracy in resume ranking due to the lack of an established method.

The current approach involves utilizing various machine learning algorithms to sort through resumes. However, many systems have struggled to effectively rank CVs. To address this issue, a stacked model is used where a series of machine earning algorithms are employed in the ranking process. By combining natural language processing techniques for text processing and machine learning for ranking, a more optimal solution can be achieved compared to existing methods.

- The unavailability of a suitable approach for providing a summary representation of skill proficiency is an issue.

While earlier studies have emphasized presenting a candidate's CV through knowledge graphs or other graphical formats, a graphical representation may be more effective in comparing a candidate's skills to job requirements. By utilizing a graphical format for this purpose, it can be easier to comprehend a candidate's skills in relation to the job requirements.

The proposed system primarily aims to rank resumes based on a generated job description, created using input from recruiters. Additionally, the system outputs a graphical representation of skill proficiency, enabling companies to gain an overview of candidates' skill level before recruiting them in the future .this research is done to provide a solution to these identified gaps by developing a system with the ability to rank resumes and identify candidate skills to get a best fit for a job opportunity.

	Research A	Research B	Research C	Research D	Developed System
Generate job-description in a proper structure	✗	✗	✗	✗	✓
Summary representation of candidate skills compared to JD	✗	✓	✗	✗	✓
Matching job description with cv	✓	✗	✗	✓	✓
Focus on IT industry job positions	✓	✗	✓	✗	✓

Table 1 Research Gap through Research Papers

1.4 Research Problem

The traditional process of manual resume screening can be a tedious and time-consuming task, especially for large organizations that receive a high volume of applications. With the increasing competition in the job market, it is essential [3] for companies to attract and retain top talent quickly and efficiently. This is where a CV ranking system comes in handy.

Recruiters may not be able to create an accurate job description that ensures proper ranking of resumes. Creating a structured job description is crucial to hiring the right candidates. Therefore, it is necessary to develop a feature that allows them to generate a structured job description.

According to a survey conducted by the research team, job descriptions are more important in ranking. However, most existing systems focus solely on ranking and do not consider the importance of a proper job description.

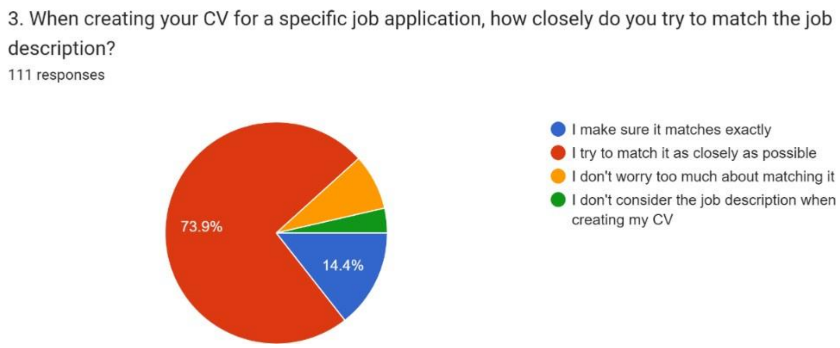


Figure 5 Candidates match CV to JD

Figure 4 shows that most of the responses prefer making resumes according to the job descriptions.

Automated CV [4] ranking systems can help companies reduce time and resources needed for candidate screening while providing a more accurate and efficient selection process. Algorithms match job requirements with candidate qualifications and eliminate human bias from the selection process, ensuring a fair and unbiased assessment of candidates. Therefore, ranking systems should place greater emphasis on the accuracy of ranking algorithms.

Ranking resumes alone is not an effective way to identify a candidate's skills. A more comprehensive overview of skills using graphical methods would provide a better understanding of a candidate's capabilities. Developing this feature gives a better explanation of each candidate’s skills.

By utilizing these two methods, recruiters and HR managers can conveniently compare the proficiency level of several candidates, enabling them to make well- informed decisions as to who is the best fit for the job.

Relying solely on resume rankings to assess a candidate's skills may not be the most effective approach. Resumes often provide a limited view of a candidate's capabilities, with qualifications and experiences condensed into a few bullet points. To gain a deeper and more comprehensive understanding of a candidate's abilities, it's advantageous to employ graphical methods that visually represent their skill set. These graphical representations can include skill matrices, proficiency charts, or visual summaries of a candidate's qualifications. Such visual aids offer a clearer picture of a candidate's strengths and weaknesses, making it easier to identify their suitability for a particular role. By incorporating these graphical tools alongside traditional resume assessments, recruiters and HR managers gain a powerful tool to evaluate candidates more holistically.

The benefit of using both methods is that it enables recruiters and HR managers to efficiently compare the skill levels of multiple candidates. This side-by-side comparison makes it simpler to assess which candidate aligns best with the job requirements. Instead of solely relying on written descriptions in resumes, which can sometimes be subjective or unclear, graphical representations provide a more objective and intuitive means of assessment. Incorporating these graphical tools into the hiring process empowers decision-makers to make well-informed choices about which candidate is the best fit for the job. It reduces the risk of overlooking potentially valuable skills and ensures that the selected candidate not only meets the basic qualifications but also possesses the specific proficiencies needed to excel in the role. Ultimately, this approach enhances the overall effectiveness and accuracy of the candidate evaluation process, leading to better hiring decisions and more successful placements.

1.5 Research Objectives

1.5.1 Main Objective

The focus of this study is to create a sophisticated and efficient system that can effectively evaluate, and rank job applicants' resumes based on the requirements outlined in the job description provided by the recruiter. The primary goal of this system is to streamline the hiring process and save time for recruiters and hiring managers by automating the initial screening process. Additionally, the proposed system will provide a comprehensive graphical representation of each candidate's skill proficiency, allowing recruiters to compare candidates' strengths and weaknesses quickly. This feature will enable recruiters to make more informed decisions and select the most suitable candidate for the job.

1.5.2 Specific Objectives

Specific objectives with the research component are:

- **Generate a proper structure for a job description.**

The proposed chatbot system is designed to streamline the job description creation process by collecting relevant information and data from recruiters to create a consistent and accurate job description format for all job openings. Based on this information, the chatbot will generate a job description in a standardized format, ensuring consistency across all job openings

- **Rank the candidate resume according to the Job description.**

In the proposed system, candidate resumes will be evaluated and ranked based on their relevance to the specific job requirements. By comparing the candidate's qualifications, experience, and skills with the job requirements, the system will assign a score to each candidate, indicating their suitability for the role.

- **Analyze the resume and graph the extracted data of cv and Job Description.**

Once the system has ranked the resumes and generated an overall skill proficiency graph for each candidate, the next step is to compare the job requirements with the candidate's skills and experience using a graphical representation.

2 METHODOLOGY

The proposed system aims to utilize resumes to rank job candidates and identify the best fit for a particular job vacancy. To accomplish this, we must first create a job description that outlines the specific requirements of the role. Next, we must extract relevant data from the candidate's resume. Once the data from both the job description and the resume have been preprocessed, a machine-learning model will be used to rank the candidates. Finally, a graphical representation will be generated to compare the skills of the top candidate.

2.1 Methodology of CV Ranking

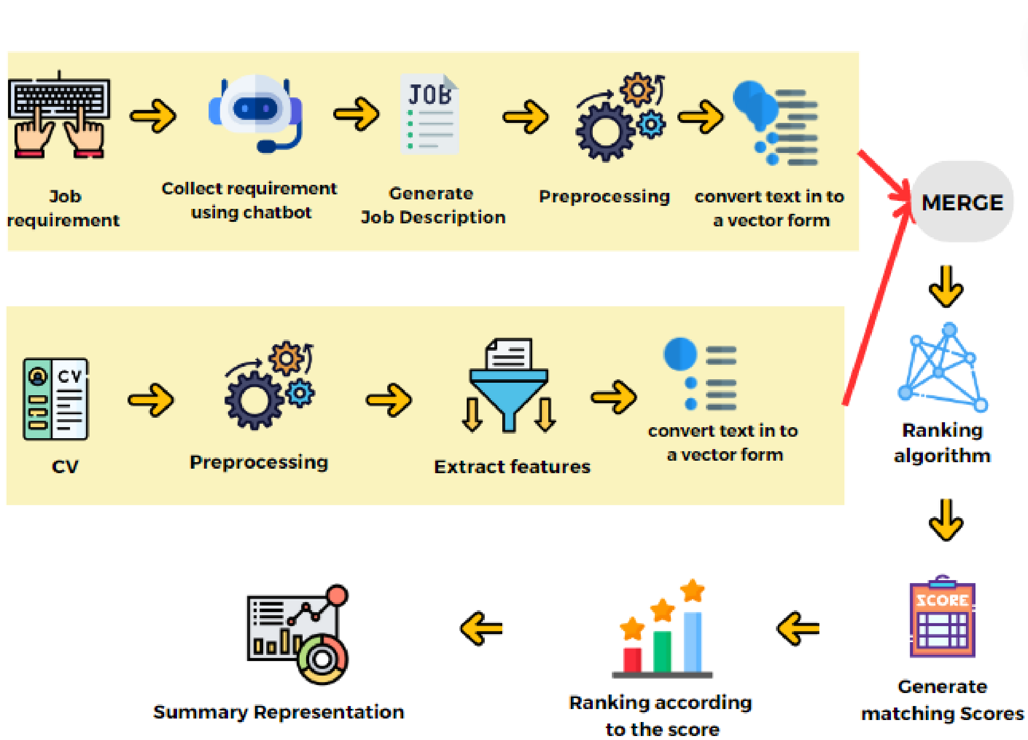


Figure 6 System Architecture of CV Ranking

2.1.1 Automated Generation of Structured Job Description

In the project's initial phase, we focused on creating a job description generator that operates within an interactive environment, using an AI-powered chatbot. This approach involved carefully gathering a wide range of prompts to guide the process of crafting well-structured job descriptions. These prompts covered essential elements like job titles, required qualifications, and specific job requirements. This effort was a significant milestone in developing an advanced tool that automates job description creation, ensuring that all critical details are included. Ultimately, this tool aims to improve the accuracy and efficiency of job posting and recruitment processes.

2.1.1.1 Architecture of Chatbot

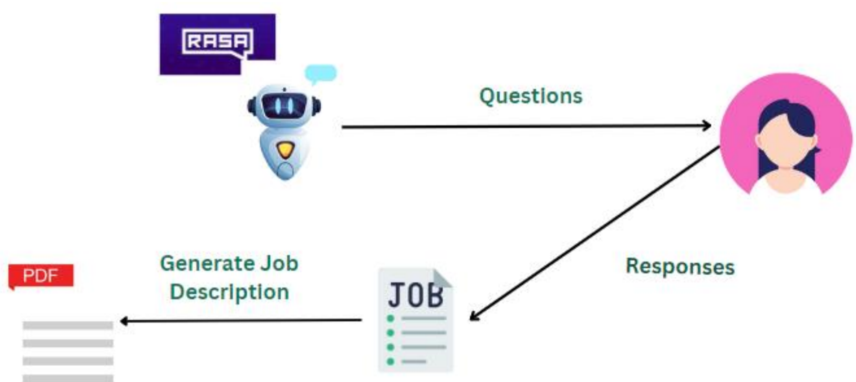


Figure 7 Architecture of JD Bot

During the job description chatbot's development, we used intent classification to train the Rasa chatbot for accurate user message understanding. As the chatbot started responding, these clear intents, supported by practical examples, played a crucial role in shaping the bot's answers. This methodical use of intents and examples provided a structured foundation that directed the chatbot's interactions, ensuring accurate and relevant responses, especially for job-related questions and sharing information. This approach emphasized the chatbot's capacity to provide customized and contextually relevant answers, enhancing the overall user experience.

```yaml
- intent: request_job_description
  examples: |
    - i want to create a job description
    - give job description
    - job description
    - how to create a job description
    - make job description
    - job
    - description

- intent: again
  examples: |
    - Start over
    - Begin again
    - Restart
    - Refresh
    - Run it one more time
```

Figure 8 Intents implementation of JD Bot

To guide the chatbot through its conversation flow and obtain the necessary responses, stories were created. In the use case of developing a job description chatbot, the aim was to prompt specific questions and capture the responses efficiently, storing them in designated slots for future reference. The actions in Rasa were implemented to both save the responses and facilitate the generation of a PDF.

```python
slot_values = [
    first_name,
    last_name,
    job_title,
    job_summery,
    job_Responsibilities,
    job_Qualifications,
    job_Education,
    job_Employment_Type,
    job_Work_Schedule,
    job_Location,
    job_Salary,
    job_Application_Process,
]

pdf_filename = "output.pdf"
doc = SimpleDocTemplate(pdf_filename, pagesize=letter)
styles = getSampleStyleSheet()
story = []
for slot_name, slot_value in zip(tracker.slots.keys(), slot_values):
    story.append(Paragraph(f"{slot_name}: {slot_value}", styles['Normal']))
doc.build(story)

dispatcher.utter_message(text=f"I've created a PDF with the slot values. ")
```

Figure 9 PDF generation of JD bot

2.1.2 Predicting matching percentage of resume and job description.

2.1.2.1 Dataset collection

A dataset containing 90 resumes along with job descriptions and matching percentages was obtained from Kaggle for the model-building process. The resumes were originally in PDF forms and were related to IT industry job positions. For matching purposes, the generated job description is used.

2.1.2.2 Data Extraction of Resumes

Since all the resumes and job descriptions are in PDF format, we utilized a PyPDF2 parser to extract the data.

```python
def pdf2Text(filename):
    ''' load pdf and return the text'''
    text = ''
    # open the pdf file
    with open(filename, 'rb') as file:
        # create a PDF reader object
        reader = PyPDF2.PdfReader(file)

        # loop over each page
        for page in reader.pages:
            # extract text from the page and concat
            text += page.extract_text()

    # return all texts
    return text
```

Figure 10 Data extraction from resumes and JD

2.1.2.3 Matching resumes with job descriptions

This component is focused on matching the extracted resume data and job description. Here the final score will be the matching percentage between resume and job description.

The dataset is initially cleaned to remove any noise.

```
# convert education degrees like B.Tech or BTech to a specified form
x = re.sub(r"\s+b[.]?[ ]?tech[(. /]{1}", " btech bachelor of technology ", x)
x = re.sub(r"\s+m[.]?[ ]?tech[(. ]{1}", " mtech master of technology ", x)
x = re.sub(r"\s+b[.]?[ ]?a[(. ]{1}", " ba bachelor of arts ", x)
x = re.sub(r"\s+m[.]?[ ]?a[(. ]{1}", " ma master of arts ", x)
x = re.sub(r"\s+b[.]?[ ]?sc[(. ]{1}", " bsc bachelor of science ", x)
x = re.sub(r"\s+m[.]?[ ]?sc[(. ]{1}", " msc master of science ", x)
x = re.sub(r"\s+b[.]?[ ]?e[(. ]{1}", " beng bachelor of engineering ", x)
x = re.sub(r"\s+m[.]?[ ]?e[(. ]{1}", " meng master of engineering ", x)
x = re.sub(r"\s+b[.]?[ ]?c[.]?[ ]?a[(. ]{1}", " bca bachelor of computer applications ", x)
x = re.sub(r"\s+m[.]?[ ]?c[.]?[ ]?a[(. ]{1}", " mca master of computer applications ", x)
x = re.sub(r"\s+b[.]?[ ]?b[.]?[ ]?a[(. ]{1}", " bba bachelor of business administration ", x)
x = re.sub(r"\s+m[.]?[ ]?b[.]?[ ]?a[(. ]{1}", " mba master of business administration ", x)
```

Figure 11 Cleaning resume data

Next the process of concatenating data was implemented.

```
def concat(s):
    '''Concatenate words like "D A T A  S C I E N C E" to get "DATA SCIENCE"'''

    # add spaces at both end for better processing

    s = ' '+s+' '

    while True:

        # search if more than two alphabets are separated by space

        x = re.search(r"(\s[a-zA-Z]){2,}\s", s)

        if x==None:

            break
        # replace to get the concatenation

        s = s.replace(x.group(),' '+x.group().replace(' ','')+' ')

    return s
```

Figure 12 Concatenation of words in resumes

Next, the text is lemmatized to build the model. The WordNetLemmatizer from the NLTK library is used for this purpose.

```
for i in x.split():
    if not (removeStopWords and i in stopwords.words('english')):
        # use lemmatizer to reduce the inflections
        lemmatizer = WordNetLemmatizer()
        i = lemmatizer.lemmatize(i)
        z.append(i)
z = ' '.join(z)

# strip white spaces
z = z.strip()
return z
```

Figure 13 Lemmatization of Resume words

2.1.2.4 Feature Extraction

The feature encoding of the resumes and job descriptions primarily focused on extracting numerical features such as common words, word counts in the resumes, and fuzz ratios. These features provided valuable information for subsequent analysis.

```
# number of words in resume
data['resume_word_num'] = data.processed_resume.apply(lambda x: len(x.split()))
# number of unique words in job description and resumes
data['total_unique_word_num'] = data.apply(lambda x: len(set(x.job_description.split()).union(set(x.processed_resume.split()))) ,axis=
# number of common words in job description and resumes
data['common_word_num'] = data.apply(lambda x: len(set(x.job_description.split()).intersection(set(x.processed_resume.split()))) ,axis
# number of common words divided by total number of unique words combined in both job description and resumes
data['common_word_ratio'] = data['common_word_num'] / data.apply(lambda x: len(set(x.job_description.split()).union(set(x.processed_re
# number of common words divided by minimum number of unique words between job description and resumes
data['common_word_ratio_min'] = data['common_word_num'] / data.apply(lambda x: min(len(set(x.job_description.split())), len(set(x.proc
# number of common words divided by maximum number of unique words between job description and resumes
data['common_word_ratio_max'] = data['common_word_num'] / data.apply(lambda x: max(len(set(x.job_description.split())), len(set(x.proc

# applying the Fuzz WRatio algorithm to calculate similarity scores between the "job_description" and "processed_resume" columns in a
data["fuzz_ratio"] = data.apply(lambda x: fuzz.WRatio(x.job_description, x.processed_resume), axis=1)
# Fuzz partial ratio
data["fuzz_partial_ratio"] = data.apply(lambda x: fuzz.partial_ratio(x.job_description, x.processed_resume), axis=1)
# Fuzz token set ratio
data["fuzz_token_set_ratio"] = data.apply(lambda x: fuzz.token_set_ratio(x.job_description, x.processed_resume), axis=1)
# Fuzz token sort ratio
data["fuzz_token_sort_ratio"] = data.apply(lambda x: fuzz.token_sort_ratio(x.job_description, x.processed_resume), axis=1)

# is fresher
data['is_fresher'] = data.processed_resume.apply(lambda x: int('fresher' in x.split()))
```

Figure 14 : Feature Extraction of resume dataset

2.1.2.5 Data splitting

```
X_train, X_test, y_train_, y_test = train_test_split(X, y, test_size=0.30, random_state=1)
```

Figure 15 Data Splitting of resume dataset

2.1.2.6 Feature Vectorization

Finally, the extracted text was vectorized under two methods. Count vectorizer and tf-idf vectorizer was used in Scikit learn for this.

1. Feature vectorization using Count Vectorizer
2. Feature vectorization using TF-IDF Vectorizer.

```
vectorizer1 = CountVectorizer()

job_transformed = vectorizer1.fit_transform(X['job_description'])
resume_transformed = vectorizer1.fit_transform(X['processed_resume'])

# Combine the transformed data horizontally
X_transformed = hstack((job_transformed, resume_transformed))
```

Figure 17 : Feature Vectorization

```
vectorizer1 = TfidfVectorizer()

job_transformed = vectorizer1.fit_transform(X['job_description'])
resume_transformed = vectorizer1.fit_transform(X['processed_resume'])

# Combine the transformed data horizontally
X_transformed = hstack((job_transformed, resume_transformed))
```

Figure 16: Combining transformations.

2.1.2.7 Model Training

Following the extraction of features, several machine learning models were employedto identify the best-performing ones among them.Following are the machine learning models used,

- Linear Regression Model
- KNN Regression Model
- Decision Tree Regression Model
- Support Vector Regression Model – Linear Kernal
- Support Vector Regression Model – RBF Kernal
- Random Forest Regression Model

To optimize performance, hyperparameter tuning was carried out using the GridSearchCV technique, enabling the identification of the best parameters and the selection of the most optimal model..

```
from sklearn.model_selection import GridSearchCV
param_grid = {
   'C': [0.1, 1, 10],
    'epsilon': [0.01, 0.1, 1]
}
grid_search =GridSearchCV(estimator=svr_model_linear ,param_grid =param_grid ,cv =5)
grid_search.fit(X_train,y_train)
best_param =grid_search.best_params_
grid_search.score(X_test,y_test)
```

Figure 18 : Hyper parameter tuning.

```
print( grid_search.score(X_train,y_train))
best_model = grid_search.best_estimator_
print(best_model)
print(best_param)
```

Figure 19 Selecting Best Model

2.1.2.8 Stacked Ensemble Model

To further enhance accuracy, the stacked method was utilized as an ensemble technique. This approach involved combining the individual models that demonstrated favorable accuracies, resulting in a single unified model with significantly improved accuracy. This methodology was applied to the data features obtained from TF-IDF vectorization and Bag-of-Words vectorization techniques.

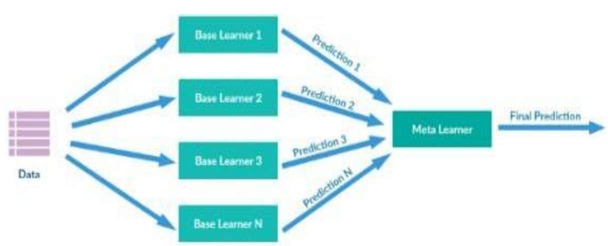


Figure 20 : Stacked Model implementations

For the implementation of the stacked model, individual models with the best accuracies were selected as the base models.

```
# Step 3: Build the Base Models
base_models = [
    ('Ridge', Ridge(alpha=0.1, fit_intercept=False, solver='lsqr')),
    ('svr_model_linear' , SVR(C=10, epsilon=0.01, kernel='linear')),
    ('rfr_model',  RandomForestRegressor(max_depth=20, n_estimators=50)),
    ('dt_model', DecisionTreeRegressor(max_depth=10, min_samples_leaf=4) ),
]

# Step 4: Build the Stacking Model
stacking_model = StackingRegressor(
    estimators=base_models,
    final_estimator = KNeighborsRegressor(metric='manhattan', n_neighbors=9, p=1, weights='distance')
)
```

Figure 21 : Stacked model codes

K-Fold cross-validation was used to assess the model's performance. The dataset was split into training and test sets in each fold.

The stacking model was trained on the training set and evaluated on the test set in each fold.

```python
kf = KFold(n_splits=5, random_state=42, shuffle=True)

mse_scores = []
for train_index, test_index in kf.split(X_transformed):
    X_train, X_test = X_transformed[train_index], X_transformed[test_index]
    y_train, y_test = y[train_index], y[test_index]

    # Train the stacking model
    stacking_model.fit(X_train, y_train)

    # Predict the matching percentages for the test set
    y_pred = stacking_model.predict(X_test)

    # Evaluate the model
    mse = mean_squared_error(y_test, y_pred)
    mse_scores.append(mse)

# Calculate the mean MSE across all folds
mean_mse = np.mean(mse_scores)

print("Mean Squared Error:", mean_mse)

# Calculate the score of the stacking model on the training data
train_score = stacking_model.score(X_train, y_train)

# Calculate the score of the stacking model on the test data
test_score = stacking_model.score(X_test, y_test)
```

Figure 22 : Stacked model evaluation

2.1.2.9 Keyword Extraction from Resumes

As the initial step the resume data was subjected to preprocessing steps. The clean text function serves the purpose of cleaning input text data by removing non-ASCII special characters. It takes a text string as input and returns a cleaned version of the text.

```
def clean_text(text):

    import re
    cleaned_data = re.sub(r'[^a-zA-Z0-9\s\/]', '', text)
    cleaned_data = cleaned_data.replace('/', ' ')
    return cleaned_data
```

Figure 23 : Resume data cleaning

Initially, the cleaned data was tokenized using the NLTK library in Python..

```
def nltk_tokenizer(text):

    nltk.download('punkt')
    from nltk import word_tokenize
    tokens = word_tokenize(text)
    #tokens = text.split()
    return tokens
```

Figure 24 Tokenization of words

Then the tokenized data feed into below methods to get token lists.

```
def nltk_pos_tag(token_list):

    nltk.download('averaged_perceptron_tagger')
    from nltk import pos_tag
    tagged_list = pos_tag(token_list)
    return tagged_list
```

Figure 25 get nltk pos tag

In the next step, stop words that were present in the tokenized words were removed.

```
def nltk_stopwords_removal(token_list):

    nltk.download('stopwords')
    from nltk.corpus import stopwords
    stop_words = set(stopwords.words('english'))
    stopwords_filtered_list = [w for w in token_list if w not in stop_words]
    return stopwords_filtered_list
```

Figure 26 Stop Word Removal

After subjecting to preprocessing of cleaning tokenization and stop word removal unique tokens were obtained. The code snippet assigns the variable 'keywords' with a filtered list of tokens from 'pos_tagged_tokens', where tokens are retained if they belong to specific grammatical categories: proper nouns (NNP), nouns (NN), verbs in non-3rd person singular present tense (VBP), and adjectives (JJ).

```
def nltk_keywords(data):

    data = clean_text(data)
    tokens = nltk_tokenizer(data)
    pos_tagged_tokens = nltk_pos_tag(tokens)
    keywords = filter_token_tag(pos_tagged_tokens, ['NNP', 'NN', 'VBP', 'JJ'])
    keywords = nltk_stopwords_removal(keywords)
    keywords = unique_tokens(keywords)
    #print('NLTK Keywords: ', keywords)
    return keywords
```

Figure 27 Keyword Extraction

Finally, the extracted keywords from a resume are compared with the keywords from a job description. The code iterates through each word in the job description (referred to as word). It checks if the current word from the job description is also present in the keywords resume list, which likely contains keywords extracted from the resume.

```
#Creating a table showing Match Result between JD and Resume
jd_keywords_in_resume_table = []
for word in keywords_jd:
    if word in keywords_resume:
        match_result = [word, 'Match']
    else:
        match_result = [word, 'No Match']
    jd_keywords_in_resume_table.append(match_result)

from tabulate import tabulate
print(f'Comparing Resume and Job Description:')
print(tabulate( jd_keywords_in_resume_table, headers=['Words', 'Keywords in JD', 'JD-Resume Match Result'], showindex='al

#Write output in a result file
write_string = 'Match Result between JD and Resume Keywords' +  '\n' + tabulate( jd_keywords_in_resume_table, headers=['S
dp.write_file( result_file_name, 'a', write_string)
```

Figure 28: Final keyword matching

2.1.3 CV Skills Matching

The skills provided by the recruiter were compared with the skills extracted from the CV. Unique skills and their frequencies were then collected to create a skills summary chart for further analysis.

```
def extract_skills_from_resume(text, skills_list):
    skills = []
    # Compile regular expressions for each skill once
    skill_patterns = [re.compile(rf'\b{re.escape(skill)}\b', re.IGNORECASE) for skill in skills_list]

    for pattern in skill_patterns:
        if pattern.search(text):
            skills.append(pattern.pattern)
```

Figure 29 Skills Analysis

```python
def get_skill_frequency(text, skills_list):
    skill_counts = Counter()

    for skill in skills_list:
        pattern = r"\b{}\b".format(re.escape(skill))
        matches = re.findall(pattern, text, re.IGNORECASE)
        skill_counts[skill] = len(matches)

    return skill_counts
```

Figure 30 Skills frequecies

2.1.3.1 Tools

Visual Studio code - Visual Studio Code [5] is a powerful and flexible IDE code editor that can be used to develop python codes due to its open-source nature and it’s a free IDE. It also includes features such as a built-in terminal and debugging tools, which can help developers to quickly identify and fix issues in their code.

PyCharm – This IDE provides essential tool for developers as allows developers to write high -quality code which is crucial in the devolopment process. PyCharm's interface uses color-coding to improve code readability and error detection by highlighting keywords, classes, and functions. It autocompletes feature is a time-saving efficiency booster, making it an essential tool for developers who aim to produce high-quality code while streamlining their coding process.

2.1.3.2 Technologies

Python - Python is a popular language for machine learning [17] because it offers several advantages. It is an easy-to-learn language, and it contains a large selection of libraries and frameworks. As it is a flexible language, it can be used for various tasks.

Scikit-learn – it’s a simple and efficient tool for predictive data analysis which is accessible to everybody and reusable in various contexts. It’s versatile and integrates well with other Python libraries.

NLTK (Natural Language Toolkit) – NLTK is a open source library for the python programming language. [18]As its containing text processing libraries for tokenization, parsing, classification which is needed for future works of prediction.

PDF parsing libraries - PyPDF2 – It is a library for Python that helps you do different things with PDF files. You can use it to get information from a PDF document.

.

NumPy - It is an open-source library used for data analysis and is used in most machine learning components for definition of predictive models and modeling to fit the data.

Rasa- Rasa is an open-source framework for building chatbots [6] that can understand and respond to text and voice-based conversations. These chatbots are capable of processing natural language input and maintaining a conversation with users while determining their next course of action.

2.2 Commercialization aspects of the product

Intellihire is a software solution developed by SMMS Software Solutions to streamline the recruitment process for IT companies, which will be developed as a web application, and can also be expanded to include mobile and desktop versions. By introducing this application "IntelliHire", companies can save significant amounts of time in the recruitment of candidates. Compared to existing applications in the market, this application offers new and advanced features to the recruiting people, which will enable them to get the maximum usage of automated recruiting processes. Also, this product includes free version with basic features and a paid version with advance features.

This application will allow companies to find the right candidates without facing the loss of resources. With the ability to effectively rank candidates based on job descriptions and a thorough analysis of their CVs, companies can streamline their recruitment process and quickly identify the most suitable candidates for their open positions. Additionally, by automating the recruitment process, recruiters can focus on other essential tasks, such as conducting interviews and onboarding new hires, thus improving overall productivity. This product focus mainly of HR professionals of the company. Ultimately, an automated recruiting application can provide companies with a competitive advantage, allowing them to attract and retain top talent while also saving time and resource es.

2.3 Testing and Implementation

2.3.1 Testing

The testing of each component was performed to ensure that every function operated correctly.
These are functionalities expected:

1. Predicting matching percentages of a candidates when resume and job description is provided.
2. Generating prompts to create a Job description in structured format.
3. Extraction of Skills present in Resume.

Test Case No	01
Description	Matching a resume with job description
Input	Input resume of a candidate Input the related job description for the position candidate applied for. Input required skills for the position.
Expected Output	Matching percentage

Actual Output	Matching Percentage Prediction: [33.82926765]
Result	Passs

Table 2 Testing matching percentage of a candidate

Test Case No	02
Description	Rank resumes of candidates according to the matching percentages
Input	Input resumes of candidates Input the related job description for the position candidate applied for. Input required skills for the position.
Expected Output	Ranked matching percentages of candidates from highest score to the lowest

Actual Output	Matching Percentage Prediction: [54.79086921] Matching Percentage Prediction: [52.04995315] Matching Percentage Prediction: [41.98185231] Matching Percentage Prediction: [25.39284305]
Result	Pass

Table 3 Testing matching percentages of candidates.

Test Case No	03
Description	Create job description for a specific job position
Input	Input responses for the prompted questions of the chatbot.

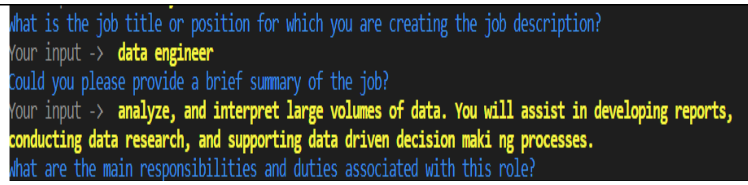
Expected Output	Job description
Actual Output	 hat is the job title or position for which you are creating the job description? our input -> data engineer ould you please provide a brief summary of the job? our input -> analyze, and interpret large volumes of data. You will assist in developing reports, conducting data research, and supporting data driven decision maki ng processes. hat are the main responsibilities and duties associated with this role?
Result	Pass

Table 4 Testing Generation of Job Description

Test Case No	04
Description	Get Matching keywords
Input	Input candidate resume Input job description
Expected Output	Matching and Not Matching Keyword List

<table>
<tr><td>Actual Output</td><td>0 | company | No Match
1 | client | No Match
2 | bods | Match
3 | etl | Match
4 | consultant | No Match
5 | sme | No Match
6 | month | Match
7 | contract | No Match
8 | basis | No Match</td></tr>
<tr><td>Result</td><td>Pass</td></tr>
</table>

Table 5 Testing Extracting Matching Keywords

Test Case No	05
Description	Match and get skills of resume
Input	Input candidate resume Input job description

Expected Output	Matching skills in resume and job description
Actual Output	{'sql': 6, 'java': 3, 'html': 2, 'python': 1, 'django':
Result	Pass

Table 6 Testing Extraction of skills

2.3.1.1 Frontend Implementation

The front-end of the application was developed using HTML and CSS, with Flask used to handle the backend logic. To enhance the visual design and user experience, Bootstrap was incorporated as a styling framework. This combination allowed for the creation of a user-friendly and aesthetically pleasing interface, with Flask handling the server-side functionalities, ensuring a seamless and responsive web application.

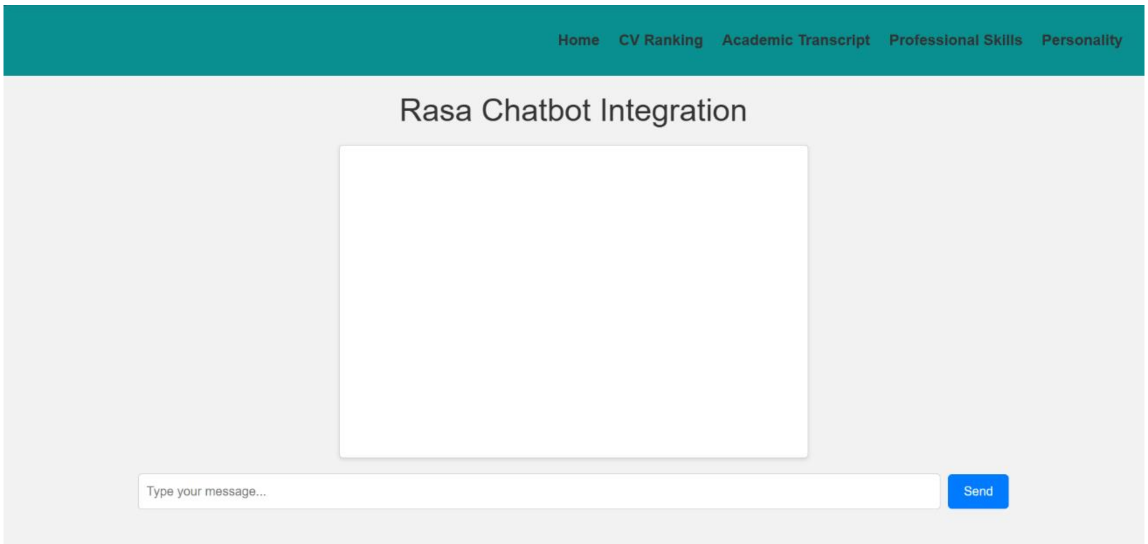


Figure 31 : Rasa chatbot User Interface

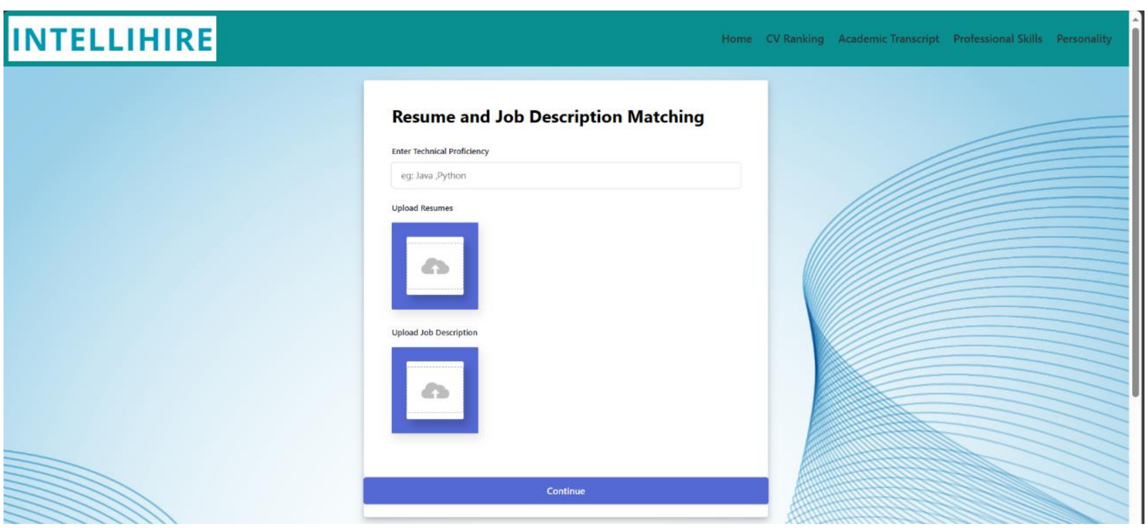


Figure 32 : Resume Uploading User Interface

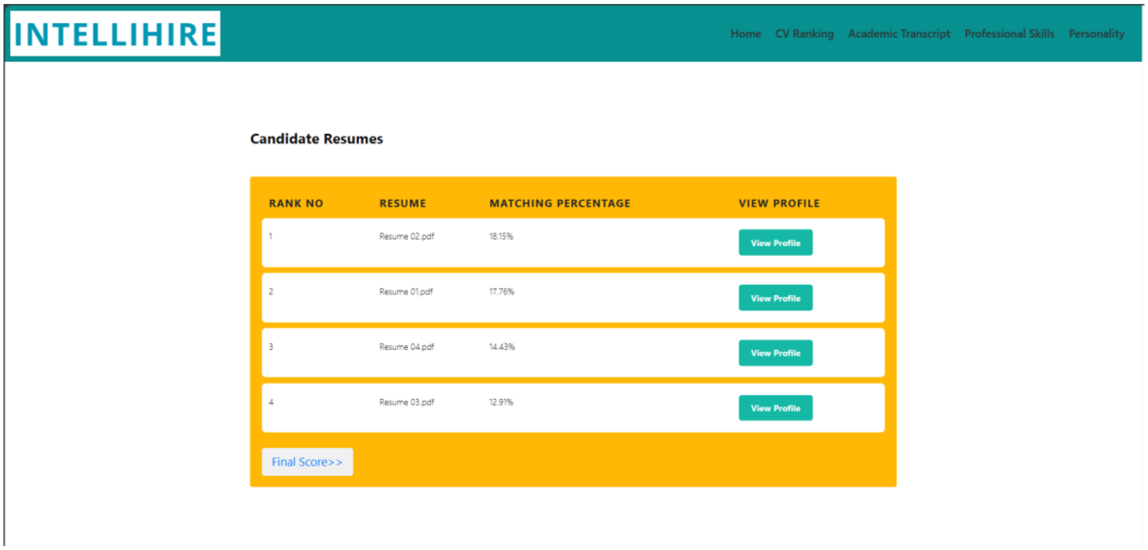


Figure 33: Resume Ranking User Interface

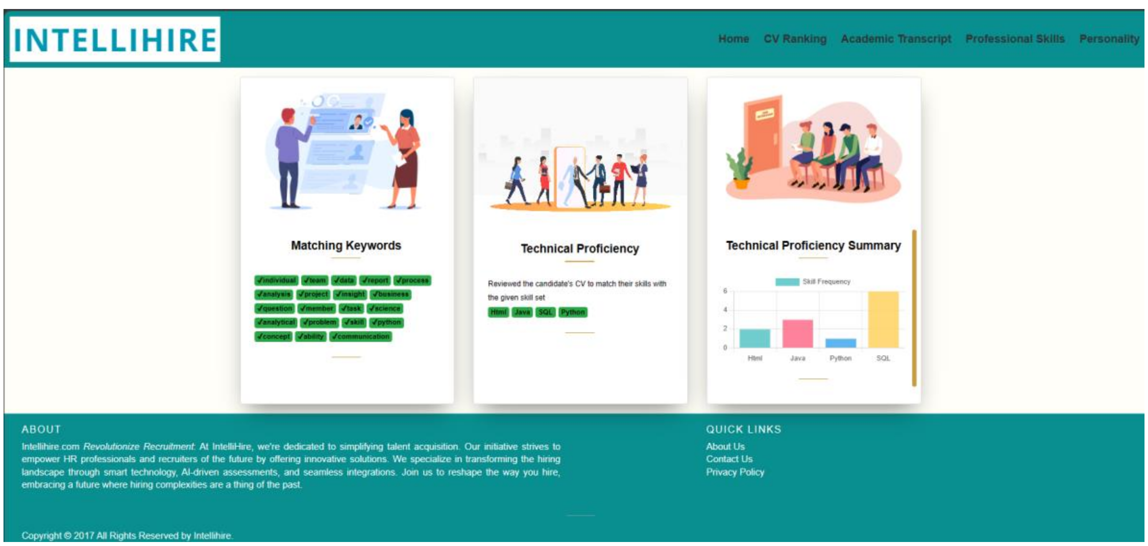


Figure 34: Candidate Skills User Interface

3 RESULTS AND DISCUSSSION

3.1 Results

The main objective of this fucntion is the choose the best candidate by analyzing cv with the job description. The following will show the results obtained .

3.1.1 Generation of Job description using a chatbot

During the development of the chatbot, a crucial step involves prompting users with questions to gather their inputs, which are then validated based on the length of the provided responses. The implementation of this methodology underscores the chatbot's commitment to quality user interactions and accurate information retrieval. By adhering to a sequential questioning strategy and employing length-based validation, the chatbot can better comprehend user requirements and provide more effective, relevant, and personalized responses.

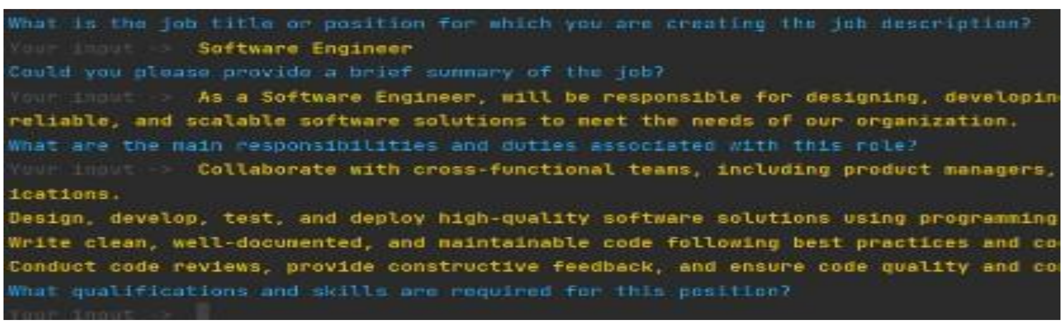


Figure 35 Chatbot JD generation

The test involved a meticulously designed set of 35 questions, strategically arranged in a sequential order to simulate a job-related inquiry. Remarkably, the chatbot demonstrated exceptional proficiency, as it accurately responded to all questions with a perfect accuracy score of 1.00.

```
rasa.core.test  - Evaluation Results on ACTION level:
rasa.core.test  -  Correct:          35 / 35
rasa.core.test  -  F1-Score:         1.000
rasa.core.test  -  Precision:        1.000
rasa.core.test  -  Accuracy:         1.000
```

Figure 36 :Chatbot accuracy for prompts

3.1.2 Predicting matching percentage of resume and job description

In the initial attempts to match resumes with job descriptions using machine learning models, the achieved accuracies fell short of expectations. Consequently, a novel approach was used by developing a stacked ensemble model that combined multiple base models for both TD-IDF and BOW vectors. Through the implementation of hyperparameter tuning and advanced cross-validation techniques, remarkable performance was accomplished.

The models and their accuracies for both TD-IDF vectors and BOW are in the table below.

Model	Accuracy before Hyper-parameter tuning	Accuracy After Hyper-parameter tuning
The Stacked Model with Count Vectorizer	0.6035	0.6810
The Stacked Model with TF-IDF Vectorizer	0.5371	0.6128

Table 7 Stacked Model accuracies

Model	Hyper parameters at the Initial Stage	Hyper parameters after Tuning
Linear Regressor	alpha=0.1	alpha=0.1, fit_intercept=False, solver='lsqr'
Support Vector Regressor RBF Kernal	kernel='linear'	C=10, epsilon=0.01, kernel='linear'
Random Forest Regressor	n_estimators=50	max_depth=20, n_estimators=50
Decision Tree Regressor	min_samples_split=10	min_samples_split=10
XGBoost	learning_rate=1	learning_rate=1, max_depth=3, n_estimators=150, reg_lambda=1000

Table 8 Hyper Parameter Tunning

3.1.3 Keyword matching and skills extraction

the resumes were matched with job descriptions based on keywords that persisted in them. The results were categorized as 'Match' and 'No Match. As a result of this approach, matching percentages based on keywords and cosine similarity were obtained for each job description.

Comparing Resume and Job Description:

Words	Keywords in JD	JD-Resume Match Result
0	company	No Match
1	client	No Match
2	bods	Match
3	etl	Match
4	consultant	No Match
5	sme	No Match
6	month	Match
7	contract	No Match
8	basis	No Match
9	role	No Match
10	time	Match
11	ir35	No Match
12	data	Match
13	team	Match
14	bau	Match
15	experience	Match
16	bo	Match
17	extract	No Match
18	transform	No Match
19	load	Match

Figure 37 : Keyword Matching of CV and JD

3.2 Research Findings

3.2.1 Job description through AI bot

When it comes to generating job descriptions, utilizing an AI-powered bot streamlines the process, ensuring that all relevant information is comprehensively included. In contrast, crafting job descriptions manually is time-consuming and challenging. Moreover, the AI chatbot can seamlessly compile the content into a professional PDF format, simplifying the task of uploading it to various professional platforms.

3.2.2 CV Ranking using Stacked Model

Based on the conducted tests and the obtained results, it became evident that the individual models were providing moderate levels of accuracy. Consequently, the next step involved the fusion of these models into a stacked ensemble. Upon thorough analysis of the outcomes, it was established that the stacked model constructed using the Count Vectorizer outperformed its counterpart, which employed the TF-IDF Vectorizer, in terms of predicting matching percentages. Furthermore, the accuracy of these models underwent enhancement through the process of hyperparameter tuning. This optimization effort significantly improved the overall performance of the models.

3.2.3 Skills Extraction from CVs

This component plays a pivotal role in extracting pertinent information from both the CV and the job description, focusing on two distinct outcomes. Firstly, it identifies matching keywords that align the CV with the requirements specified in the job description. This prioritizes CVs that closely match the job's criteria.

Secondly, the component identifies skills mentioned in the CV, validating them against the skills sought by the recruiter. This validation process ensures that the skills highlighted in the CV align with the job requirements. Additionally, these skills are cross-validated with data obtained from another component, enhancing the accuracy and reliability of the skill-matching process.

3.3 Discussion

In conclusion, the approach to resume ranking using supervised learning has demonstrated significant advancements in accuracy and effectiveness. Initially, feature extraction was employed to highlight the most salient characteristics within resumes. However, it became evident that the results obtained from individual models alone were insufficient. To enhance predictive accuracy, a stacked model was introduced, combining the strengths of these individual models. Notably, this research surpasses previous studies by leveraging a combination of regression models, which has proven to yield superior results in predicting matching percentages.

Furthermore, this research introduces a novel dimension by structuring job descriptions, simplifying the recruitment process for organizations. This structured format streamlines the matching process by targeting specific keywords, ensuring a more precise and efficient candidate selection. Additionally, by extracting candidate skills, a comprehensive analysis is provided, presented through a user-friendly interface, offering skill summarization for a more intuitive evaluation.

This research not only advances the field of resume ranking but also enhances the recruitment process through innovation and improved accuracy, ultimately benefiting both employers and candidates alike.

4 CONCLUTION

In summary, the process of accessing candidates through CVs, especially in the dynamic field of IT, has undergone a remarkable transformation. This evolution has been fueled by the integration of cutting-edge technologies and innovative methodologies. The incorporation of machine learning, natural language processing, and data-driven approaches has fundamentally reshaped the way we evaluate candidates, resulting in a more precise and efficient selection process.

This transformative shift not only accelerates and enhances the recruitment process but also deepens our insights into candidates' technical and professional capabilities. Moreover, the emphasis on concept-based categorization and tailor-made ranking techniques has enabled us to better align candidate profiles with specific job descriptions.

As a consequence, this revolution in CV assessment is not just about efficiency; it's about the strategic selection of the most fitting candidates for IT roles. In the ever-evolving realm of talent acquisition, these advancements are continuously shaping the future of candidate evaluation through CVs within the IT sector. This, in turn, leads to improved matches and serves as a catalyst for organizational growth and innovation. It represents a promising and transformative journey in the world of talent acquisition, promising to yield even more benefits in the years to come.

5 REFERENCES

[1] "83 Must-Know Resume Statistics: 2023 Data on Length, Cover Letters & Valuable Skills." https://financesonline.com/resume-statistics

[2] "Employee Selection - Purpose of selection process,"." https://www.whatishumanresource.com/employee-selection

[3] R. Day, "THE 10 MOST IMPORTANT IT SKILLS FOR 2020," Aug. 17, 2020. https://www.globalknowledge.com/us-en/resources/resource-library/articles/the-10-most-important-it-skills-for-2020/#gref

[4] T C Senanayake , G A I Limesha ,T S S Madhumali , W P I Mihirani, , "Automated CV Analyzing and Ranking Tool to Select Candidates for Job Positions," in ICIT 2018: Proceedings of the 6th International Conference on Information Technology: IoT

[5] Rohini Nimbekar ,Y ogeshP atil ,RahulPrabhu , ShainilaMulla, "Automated Resume Evaluation System using NLP," Mumbai, India..

[6] Rasika Ransing , Akshaya Mohan, Nikita Bhrugumaharshi Emberi , Kailas Mahavarkar, "Screening and Ranking Resumes using Stacked Model," in 2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT) , 2021.

[7] yishathahira C H , Sreejith C ,Raseek C, "Combination of Neural Networks and Conditional Random Fields for Efficient Resume Parsing," in 2018 International CET Conference on Control, Communication, and Computing (IC4), Trivandrum, 2019.

[8] Abeer Zaroor , Mohammed Maree , Muath Sabha, "JRC: A Job Post and Resume Classification System for Online Recruitment," in 2017 International Conference on Tools with Artificial Intelligence, Jenin, Palestine, 2017.

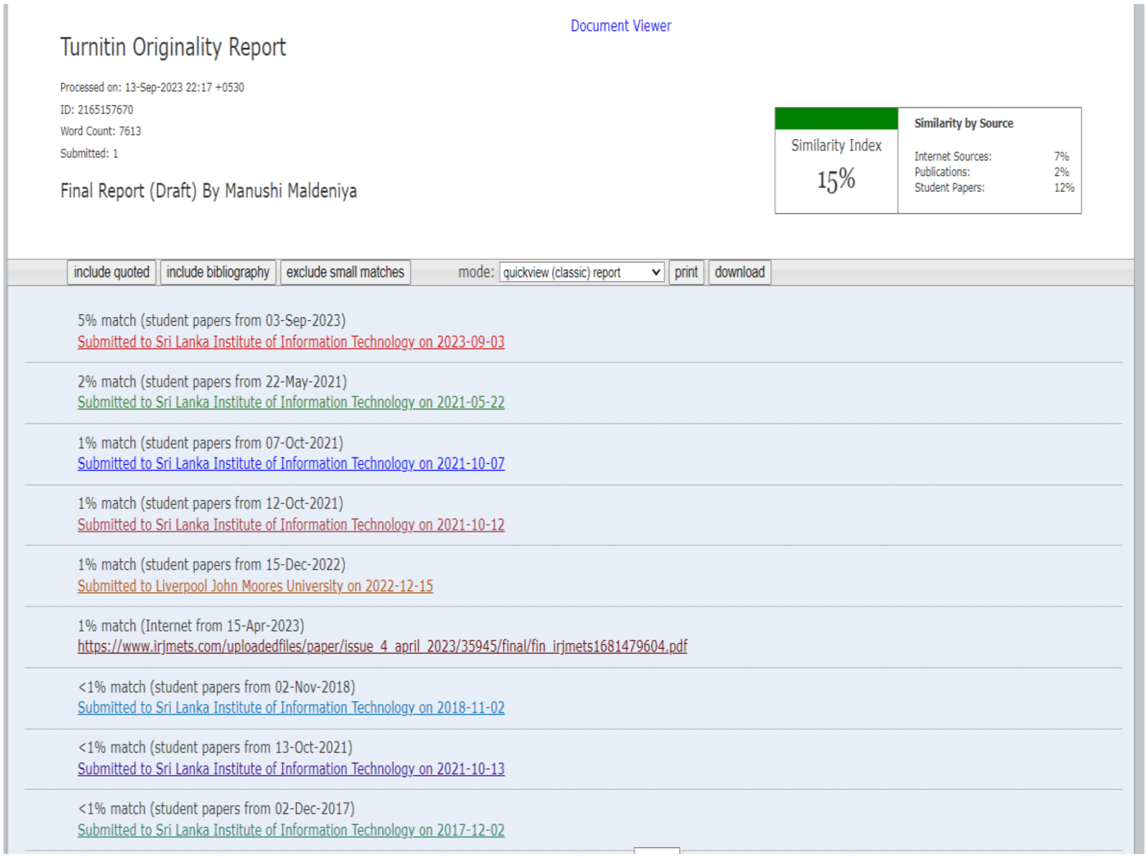
[9] Yan Wang, Yacine Allouache ,Christian Joubert, "Analysing CV Corpus for Finding Suitable Candidates using Knowledge Graph and BERT," in The

Thirteenth International Conference on Advances in Databases, Knowledge, and Data Applications, Valencia,Spain, 2021.

[10] Muntaha Mehboob , M.Saad Ali ,Saif ul Islam , Syed Sarmad Ali, " Evaluating Automatic CV Shortlisting Tool For Job Recruitment Based On Machine Learning Techniques," in 2022 Mohammad Ali Jinnah University International Conference on Computing (MAJICC), 2022.

[11] Muskan Sharma ,Gargi Choudhary , Seba Susan, "Resume Classification using Elite Bag-of-Words Approach," in Proceedings of the 5th International Conference on Smart Systems and Inventive Technology (ICSSIT 2023), Delhi, India, 2023.

Appendix : Turnitin Report



Document Viewer

Turnitin Originality Report

Processed on: 13-Sep-2023 22:17 +0530
ID: 2165157670
Word Count: 7613
Submitted: 1

Final Report (Draft) By Manushi Maldeniya

Similarity Index	Similarity by Source	
15%	Internet Sources:	7%
	Publications:	2%
	Student Papers:	12%

include quoted | include bibliography | exclude small matches | mode: quickview (classic) report | print | download

5% match (student papers from 03-Sep-2023)
Submitted to Sri Lanka Institute of Information Technology on 2023-09-03

2% match (student papers from 22-May-2021)
Submitted to Sri Lanka Institute of Information Technology on 2021-05-22

1% match (student papers from 07-Oct-2021)
Submitted to Sri Lanka Institute of Information Technology on 2021-10-07

1% match (student papers from 12-Oct-2021)
Submitted to Sri Lanka Institute of Information Technology on 2021-10-12

1% match (student papers from 15-Dec-2022)
Submitted to Liverpool John Moores University on 2022-12-15

1% match (Internet from 15-Apr-2023)
https://www.irjmets.com/uploadedfiles/paper/issue_4_april_2023/35945/final/fin_irjmets1681479604.pdf

<1% match (student papers from 02-Nov-2018)
Submitted to Sri Lanka Institute of Information Technology on 2018-11-02

<1% match (student papers from 13-Oct-2021)
Submitted to Sri Lanka Institute of Information Technology on 2021-10-13

<1% match (student papers from 02-Dec-2017)
Submitted to Sri Lanka Institute of Information Technology on 2017-12-02