TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

KALANKI, KATHMANDU

[CT654]

A MINOR PROJECT REPORT ON
"AUTOMATED RESUME SCREENING SYSTEM"

Submitted by:

Dev Kumar Karmacharya	(ACE078BCT025)
Elis Poudel	(ACE078BCT028)
Joshan Adhikari	(ACE078BCT031)

Project Supervisor:

Er. Laxmi Prasad Bhatt

Er. Dhiraj Pyakurel

A Minor project Final report submitted to the department of Electronics and Computer Engineering in the partial fulfillment of the requirements for degree of Bachelor of Engineering in Computer Engineering

Kathmandu, Nepal

March 7, 2025

AUTOMATED RESUME SCREENING SYSTEM

Submitted by:

Dev Kumar Karmacharya	(ACE078BCT025)
Elis Poudel	(ACE078BCT028)
Joshan Adhikari	(ACE078BCT031)

Supervised by:

Er. Laxmi Prasad Bhatt

Er. Dhiraj Pyakurel

A MINOR PROJECT SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER OR ELECTRONICS & COMMUNICATION ENGINEERING

Submitted to:

"DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING"

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

Balkhu, Kathmandu

March 7, 2025

LETTER OF APPROVAL

The undersigned certify that they have read and recommended to the Institute of Engineering for acceptance, a project report entitled "**Automated Resume Screening System"** submitted by:

Dev Kumar Karmacharya	(ACE078BCT025)
Elis Poudel	(ACE078BCT028)
Joshan Adhikari	(ACE078BCT031)

In the partial fulfillment of the requirements for the degree of Bachelor's Degree in Computer Engineering.

…………………………..

Project Supervisor

Er. Laxmi Prasad Bhatt

Academic Project Coordinator

Department of Electronics and Computer Engineering

…………………………..

Project Supervisor

Er. Dhiraj Pyakurel

Deputy Head of Department of Electronics and Computer Engineering

Department of Electronics and Computer Engineering

…………………………..

Academic Project Coordinator

Er. Laxmi Prasad Bhatt

Department of Electronics and Computer Engineering

March 7, 2025

COPYRIGHT

Head of Department
Department of Electronics and Computer Engineering
Advanced College of Engineering and Management
Balkhu, Kathmandu
Nepal

ACKNOWLEDGEMENT

We would like to express our profound gratitude and deep regards to our respected supervisor Er. Laxmi Prasad Bhatt and Er. Dhiraj Pyakurel, for his insightful advice, motivating suggestions, invaluable guidance, help, and support in the successful completion of this project and also for his/her constant encouragement and advice throughout our Bachelor's program.

We express our deep gratitude to **Er. Prem Chandra Roy**, Head of the Department of Electronics and Computer Engineering, **Er. Dhiraj Pyakurel** Deputy Head, Department Of Electronics and Computer Engineering, **Er. Laxmi Prasad Bhatt,** Academic Project Coordinator, Department Of Electronics and Computer Engineering, for their support, co-operation, and coordination.

The in-time facilities provided by the department throughout the Bachelor's program are also equally acknowledgeable.

We would like to convey our thanks to the teaching and non-teaching staff of the Department of Electronics & Communication and Computer Engineering, ACEM for their invaluable help and support throughout Bachelor’s Degree. We are also grateful to all our classmates for their help, encouragement, and invaluable suggestions.

Finally, yet more importantly, We would like to express our deep appreciation to our grandparents, parents, and siblings for their perpetual support and encouragement throughout the Bachelor’s degree period.

Project Members:

Dev Kumar Karmacharya (ACE078BCT025)

Elis Poudel (ACE078BCT028)

Joshan Adhikari (ACE078BCT031)

ABSTRACT

Volume resume screening of any company, which, when done manually, is ineffective, inconsistent, and time-consuming. Traditional resume screening techniques comprise filtering and shortlisting of applicants by human resource professionals themselves, which, subsequently, delays the hiring process and sometimes lets promising candidates slip through. Since the volume of job applicants is growing day by day, companies need some automation that can help the candidate selection process be efficient and faster. To manage these issues, we suggest an Automated Resume Screening System to assist HR professionals in the rapid as well as accurate identification of the best candidates for a provided job description.

It relies on Natural Language Processing (NLP) and Machine Learning algorithms to use, extract, and rank resumes that fit the job description. The system extracts the text from resumes, both in PDF and DOCX formats, and pre-processes the text by removing unwanted characters and normalizing it. The system also conducts a comparison of the extracted information and job descriptions on the basis of semantic similarity approaches such as word embedding and context matching models such as BERT. The system gives a similarity score for each resume that will enable HR professionals to filter and shortlist candidates very fast and easily on the basis of relevance. This system reduces hiring time, minimizes human error, and optimizes the candidate selection process through the automation of resume screening. It saves HR teams the hassle of manually going through hundreds of resumes, as the ranked shortlist of candidates would be presented by the system itself. The AI-based solution also guarantees that the system matches resumes contextually and not just by the occurrence of keywords, thereby making the shortlisting of candidates more relevant and accurate.

Implementation of the system validates that automated resume screening is significantly quicker, scalable, and consistent than manual filtering. The system will further be augmented by integrating it with HR management software, increasing contextual analysis in job-role matching, and making it compatible with different resume formats.

Keywords*: Artificial Intelligence, Hiring Automation, Machine Learning, Natural Language Processing, Resume Screening, Recruitment Efficiency, Job Matching*

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CV	Curriculum Vitae
DFD	Data Flow Diagram
ER	Entity-Relationship
GDPR	General Data Protection Regulation
HR	Human Resources
ML	Machine Learning
NER	Named Entity Recognition
NLP	Natural Language Processing

CHAPTER 1

INTRODUCTION

The process of recruiting in companies has undergone a rapid change in job placement which was caused by internet connectivity. It made the recruiting process for many organization to identify top talent and molded the recruiting process according to their specific business goals. In this current world where companies receive enormous volumes of resumes, sometimes even more than a thousand for a single job position. Manually going through resumes is a tiring, error-prone, and often biased process. For solving these problems, we designed Automated Resume Screening System a platform solution that manages and serves for HR teams.

1.1 Background

Looking back in couple of years, we can notice technological evolution in the every field like professionals, among which AI and NLP are one of them. These techniques have helped recruiters to execute automatic and faster screenings. Automated resume screening tools that are used by many firms and companies to reduce the time spent on the initial candidate evaluations have become a key method in their quest to be more efficient. Looking back the old trend where HR teams used to manually shortlist the candidate which is time consuming and error-prone and most likely to overlook the best candidate suitable for the required job. Our system can solve this problem by shortlisting the candidates and ranking based on the score provided from our automatic system. Also it can save the time of HR team allowing them to focus on more important task like interview. Overall, our system would reduce the hiring time, enhance the user Experience, and manage a fair and efficient talent acquisition, too.

1.2 Motivation

The motive behind this systems is to create a smart, efficient, and fair recruitment ecosystem that do good both HR teams and job seekers. By reducing inefficiency and enhancing the user experience, this system purpose to revitalize traditional hiring exercise and align with the demands of a modern.

1.3 Statement of the Problem

The hiring process plays a crucial role in talent acquisition, but it is often difficult by inefficiencies and lack of transparency. Existing analysis of system of resume screening is manual efforts that time consuming, error prone and cannot manage massive volumes of applications successfully and precisely. Resulting in the miss of qualified candidates or taking too long to place someone in a position and also the job seekers do not get good job openings. Thus to improving scalability and transparency in the recruitment process this project aims to develop an Automated Resume Screening System to resolve these issues and create a fair, efficient, and user-friendly recruitment system for both HR teams and job seekers.

1.4 Project objective

The primary objective of this project is to design and develop an Automated Resume Screening System that offers a solution for HR teams.

1.5 Significance of the study

The importance of the study is that it can change the process of recruitment. A way to bridge major inefficiencies and challenges in job seekers and HR process. For job seekers, it shall have a system that is intuitive and can offer actual transparency, easing job discovery and letting a candidate track his application process. For HR teams, it shall deploy AI and NLP to extract useful pieces of information and be more accurate in matching job descriptions with applicants. The study integrates the two platforms into one system, therefore, it enhance communication and ensure the hiring process to run smoothly and efficiently.

CHAPTER 2

LITERATURE REVIEW

Literature review is a summary of published work or simply research. It is a crucial step to understand existing work and identify gaps in the body of knowledge before conducting new research. This chapter discusses foundational studies, key technologies, challenges, and emerging trends in the field of automatic CV screening. The aim is to situate the current research within the context of existing work and highlight areas where contributions can be made.

2.1 Evolution of Automatic CV Screening

The automatic screening of curricula vitae has undergone considerable development from its inception. Early systems were designed based on keyword-based techniques, while later systems adopted more complex approaches using machine learning and NLP. When ranking a CV, different cases are taken into consideration: skills, experience, education, and location. The ranks are then used to find an appropriate match of employers and employees with the use of Gale–Shapley algorithm which eases companies for higher best possible candidates (pudasaini, shushanta, et al., 2022) [1].

To make machines figure out the similarity between documents we need to define a way to measure the similarity mathematically and it should be comparable so that machine can tell us which documents are most similar or which are least. We also need to represent text from documents in a quantifiable form (or a mathematical object, which is usually a vector form), so that we can perform similarity calculations on top of it (varun, 2020) [2].

2.1.1 Early Methods of CV Screening

Many job portals and external websites came up to reduce the difficulty of handling unstructured and diverse resumes. These require candidates to manually fill up all the information of their resume in an online form in a structured manner, thus creating a candidate metadata. The problem with this approach is that it requires redundant efforts on the part of the candidates, and they often miss out on filling complete information in these templates (Daryani, Chirag, et al., 2020) [3].

2.1.2 Recent Trends in CV Screening

In recent times, with the help of AI and NLP, modern systems are more into the contextual understanding of CVs. For instance, some NLP models, such as BERT, allow semantic matching where phrases like "managed a team" and "led a group" would be understood as similar. Machine learning models further predict candidate suitability by analyzing historical recruitment data and tailor recommendations accordingly.

2.2 Challenges and Emerging Trends

While automatic CV screening systems have shown immense promise, a number of challenges are still there, including bias, unstructured formats, and privacy concerns too. In the field of machine learning, representing text documents numerically is a challenging task. However, it is essential for various applications, such as document retrieval, web search, spam filtering, and topic modeling. Doc2Vec, a variation of the Word2Vec algorithm, provides a solution by generating vector representations from words (Dang, Kiel, 20223) [4].

2.2.1 Challenges in Automatic CV Screening

The major challenges are algorithmic bias, among others. Studies shown that biases in training data result in discriminatory outcomes; for example, the 2018 AI tool used for recruiting at Amazon was found to punish CVs containing the word "women" unfairly. Besides, unstandardized formats of CVs and compliance with data privacy laws like GDPR further complicate the implementation of CV screening systems (lriondo, roberto, 2018) [5].

2.2.2 Emerging Trends in Automatic CV Screening

To address these challenges, researchers are exploring bias detection and mitigation techniques, the development of explainable AI, and multi-modal data analysis that incorporates video resumes, portfolios, and social media profiles. Integration with end-to-end recruitment platforms and improving candidate scoring techniques are also key areas of focus.

Table 2.1: Summarized Literature Review

Year	Publication	Title	Author	Objective
2021	Lecture Notes in Networks and Systems	Rank-Based CV Screening Using the Gale–Shapley Algorithm	Pudasaini, Sushanta, et al.	Demonstrates a ranking mechanism and applies Gale–Shapley for stable matching
2020	Medium	Calculating Document Similarities Using BERT, Word2Vec, and Other Models	Varun	Demonstrates vectorizing text and measuring similarity for better CV matching
2020	Ethics and Information Technology	AN AUTOMATED RESUME SCREENING SYSTEM USING NATURAL LANGUAGE PROCESSING AND SIMILARITY	Daryani, Chirag, et al.	Presents an NLP-based solution for resume data extraction and automated screening.
2023	Medium	Job — Resume Matching — Part 1/2: Obtaining Similarity Score Using Doc2Vec	Dang, Kiel	Explains Doc2Vec-based approach for vectorizing and scoring job-resume compatibility.
2018	Machine Learning | Carnegie Mellon University	Amazon Scraps Secret AI Recruiting Engine that Showed Biases Against Women	Iriondo, Roberto	Highlights bias arising from skewed training data in an AI-based recruiting system.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1. Software Requirements

Our proposed system requires the following software's: -

i. Python

Python is high-level programming language for general-purpose programming. Python emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming in both large and small scales. Python features an automatic memory management and dynamic type system. It supports multiple programming models, including object- oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

ii. Visual Studio Code

Visual Studio is an Integrated Development Environment from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.

iii. NumPy

Numerical Python is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of highlevel mathematical functions to operate on these arrays.

iv. Spacy

spaCy is a powerful and efficient open-source library for Natural Language Processing (NLP) in Python, used for tasks such as tokenization, named entity recognition (NER), part-of-speech tagging, dependency parsing, and text classification. It offers pre-trained models that provide high-quality performance for text analysis tasks. The core **spaCy** library enables seamless text processing, while specific features like **tokenizer** break down text into manageable components such as words and punctuation. Its **NER** functionality helps identify and categorize entities like names, dates, and organizations, which is crucial for parsing resumes.

3.2 Functional Requirements

Some functional requirements of the proposed system are:-

HR Registration and Login:

Security to allow recruiters to register, login and manage their accounts.

- **Resume Parsing:**
  Extract relevant details from resumes using resume parsing automation e.g. skills, education, work experience.
- **Candidate Shortlisting:**
  Allow HR teams to filter and shortlist applicants based on pre-defined score
- **Candidate Matching:**
  AI & NLP algorithms match resumes to job descriptions.

3.3 Non-Functional Requirement

The system performance and quality attributes are defined by these requirements.

- **Scalability:**
  It will have to handle a lot of resumes and job postings as well as multiple users at the same time.
- **Performance:**
  The recommendation system should process resumes and make suggestions with extended latency.
- **Reliability:**
  Have high uptime and fault tolerance to keep services running.
- **Usability:**
  For instance, the HR interface should be intuitive, accessible, and easy for HR teams to navigate.
- **Security:**
  Use strong authentication and data protection measures to protect user data.
- **Maintainability:**
  But you want to make sure that the system is easily updatable and maintainable for future usage.

CHAPTER 4

SYSTEM DESIGN AND ARCHITECTURE

Resume sample of pdf are collected and the received pdf are converted into text format using PyPDF2. The converted texts are then filtered and is compared with important keywords which are listed in job description using Semantic Analysis.

4.1 System Flow Chart

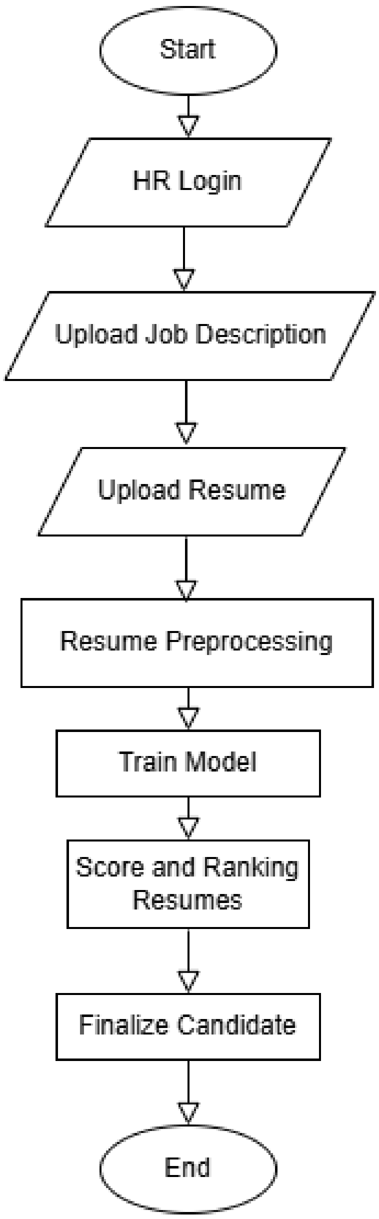

Figure 4.1: System Flow Chart

4.2 DFD Diagram

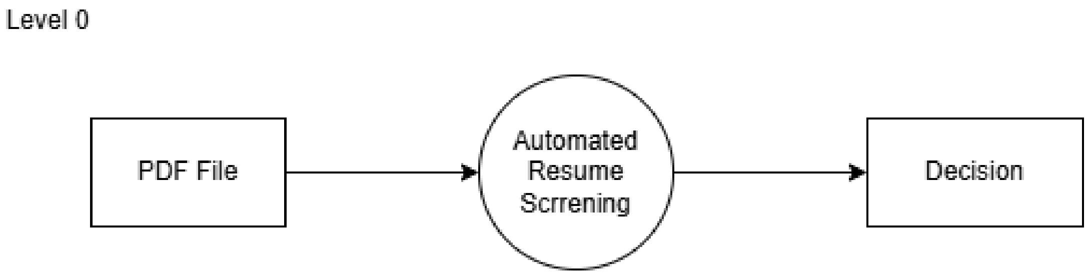

Figure 4.2.1: DFD level 0

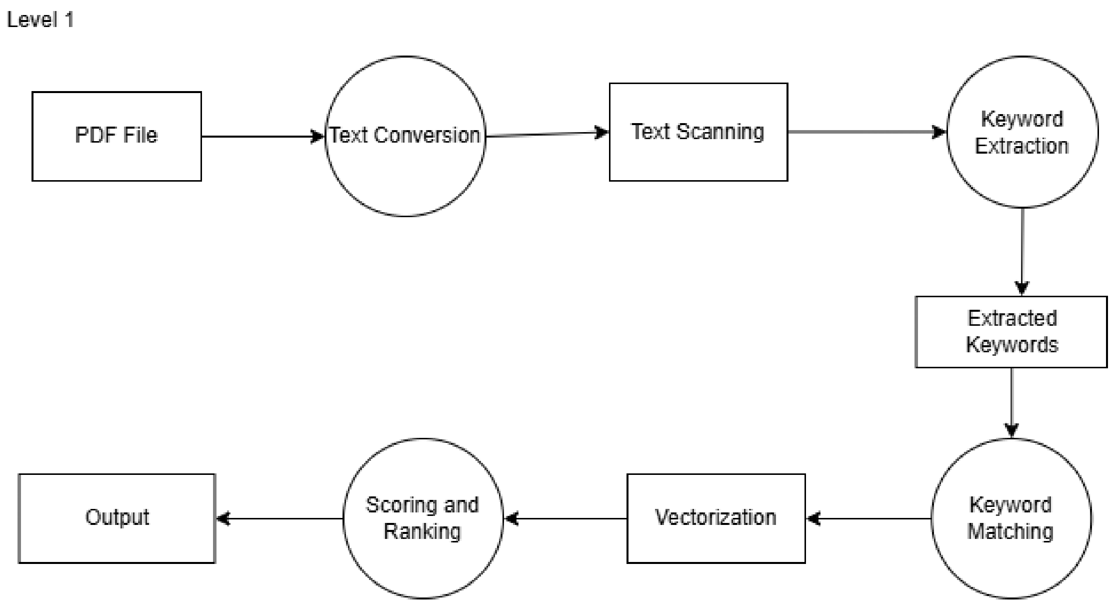

Figure 4.2.2: DFD level 1

4.3 Use Case Diagram

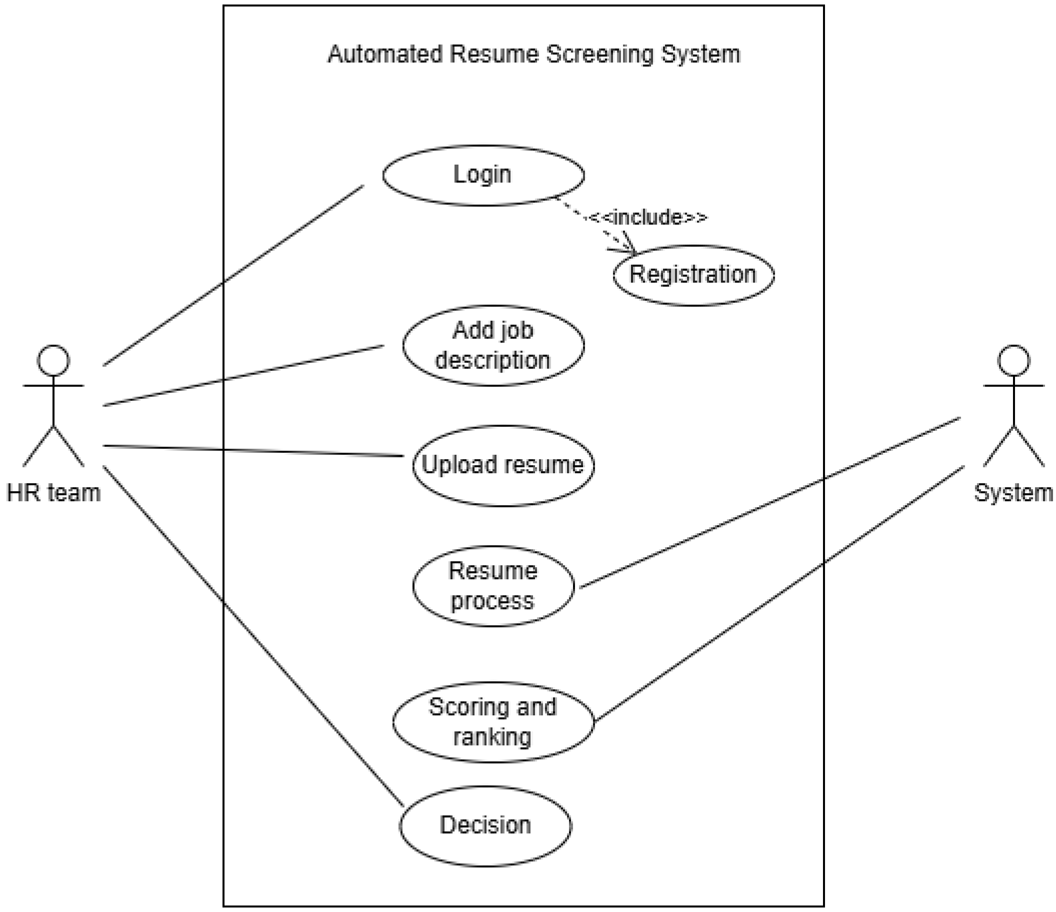

Figure 4.3: Use Case Diagram

CHAPTER 5

METHODOLOGY

The overall steps involved in designing the automated resume classification and segmentation model are as follows:

5.1 System Diagram

Our approach to solve the problem is divided it into five steps as shown in the diagram below.

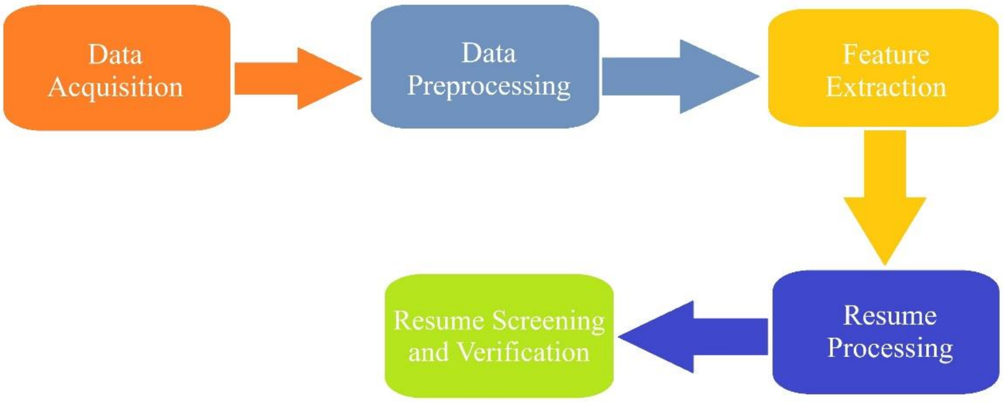

Figure 5.1: System Diagram

5.1.1 Data Acquisition

Data collection is the first step in an automated resume screening system. It will collect resumes from various sources. These resumes are available in PDF, Docx, and other formats. The main sources include online job portals, where applicants upload their resumes. Where applicants submit their CVs directly to recruiters. And some resume samples are also used to train and test the system during development.

5.1.2 Data Preprocessing

To ensure that resume checking is accurate and efficient. The pre-processing step involves standardizing and preparing the resume. The following preprocessing tasks will be performed.

- Extract content from PDF, DOCX, or TXT formats.
- Delete characters, punctuation marks. and unwanted formatting
- Split text into meaningful words or tokens.
- Eliminate common words (such as "the," "is") that don't add value.

5.1.3 Feature Extraction

Key qualifications are extracted from the resume to identify important details for screening using spacy with NER like Personal information, Skills and keywords, and Experience.

5.1.4 Resume Processing

The extracted features are analyzed using techniques from natural language processing (NLP) to understand the text and machine learning (ML) to help the system learn and make decisions.

- Keyword matching: Compare the extracted keywords to the job description using BERT.

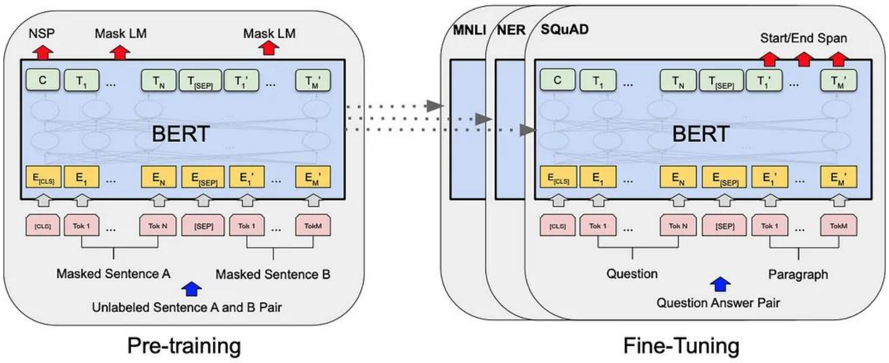

Figure 5.1.4: BERT training architecture

[BERT training architecture (Image from https://arxiv.org/pdf/1810.04805.pdf) [6]]

- Rating system: Resumes are graded based on their relevance to the workplace using cosine-similarity which is the cosine of the angle between two vectors, which gives us the angular distance between the vectors. Formula to calculate cosine similarity between two vectors A and B is is shown below (Sidrov, Grigori, etal.,2014) [7] in equation (1).

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}} \tag{1}$$

5.1.5 Resume Screening and Verification

In this step, keywords and skills extracted from each resume are ranked to match the requirements listed in the job description. Focusing on the resumes that most meet the criteria. Resumes are scored based on relevance with higher scores indicating a better fit for the job role. The review process ensures that only the most qualified resumes are considered. This will help improve the recruitment process.

5.1.6 Graphical User Interface (GUI)

We built a simple GUI as shown in figure to upload the resume image and finally screening the resume according to the job description.

5.2 Developmental Model

It is necessary to collect all necessary data and other information before developing the software. So, to develop any software modeling is one of the important phases of this cycle. According to the user interaction with the software development process and the necessary data and information we found that the Agile model is one of the best models to develop this product. As Agile model is an iterative and flexible approach to software development that emphasizes adaptability, feedback.

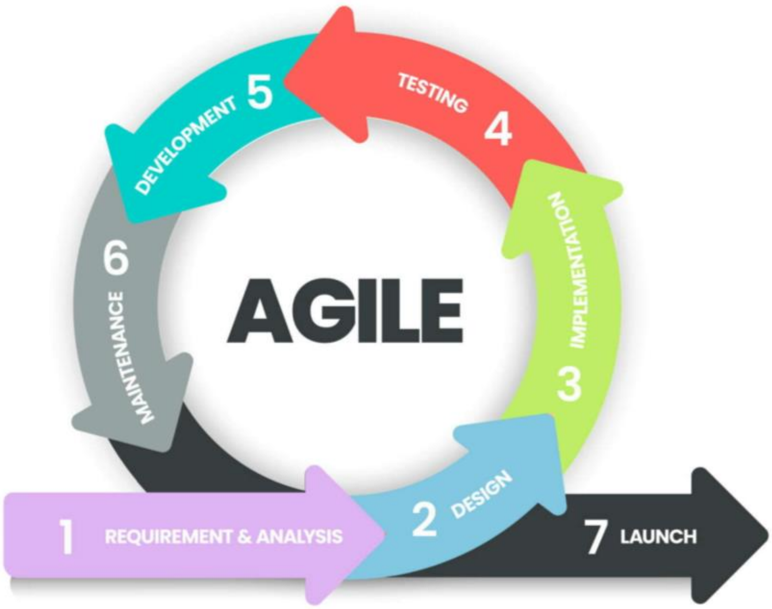

Figure 5.2: Agile model

[Source:https://proteanstudios.com/the-agile-model-and-its-application-in-software-development[8]]

CHAPTER 6

RESULTS AND ANALYSIS

6.1 RESULTS

1. Performance metrics

Our system turned out to be pretty effective as there was 80% match when we compared its top candidate picks to those chosen by human HR. This shows that our system is good at spotting the right people for the job.

2. Execution time

Our system saves time for HR teams as it cuts down time spent for screening resumes by about 70%. This can be helpful for Hr team to focus on other important parts of hiring like interviews and onboarding candidates.

3. Ranks and scoring

Our system compares and accurately ranks the resumes based on their relevance to the job description. This allows HR to know how fit the candidates are for the job.

4. Accuracy

At first we used the Distil-BERT model which was smaller, faster but gave us low accuracy so we changed to S-BERT which is found to be more accurate and reliable, which led our system to be more accurate by about 80%.

6.2 ANALYSIS

1. Efficiency

The system is able to handle large volume of resume without compromising accuracy which saves tons of time for HR teams and allow them to focus on more important aspects of the interview making it suitable for different scale organizations.

2. Accuracy and Bias mitigation

By using advanced tech like natural language processing and machine learning algorithms (like BERT), the system understands the context of resumes better than just looking for keywords. This leads to more accurate matching with job descriptions. We can keep training and updating the system to reduce any biases and make sure the selection process is fair.

3. User friendly

The user interface is designed to be really user-friendly. HR teams can upload resumes and job descriptions easily, which helps with getting everyone on board with using the system. Job seekers also benefit because their applications are processed faster and they get quicker responses.

6.2.1 Login Page

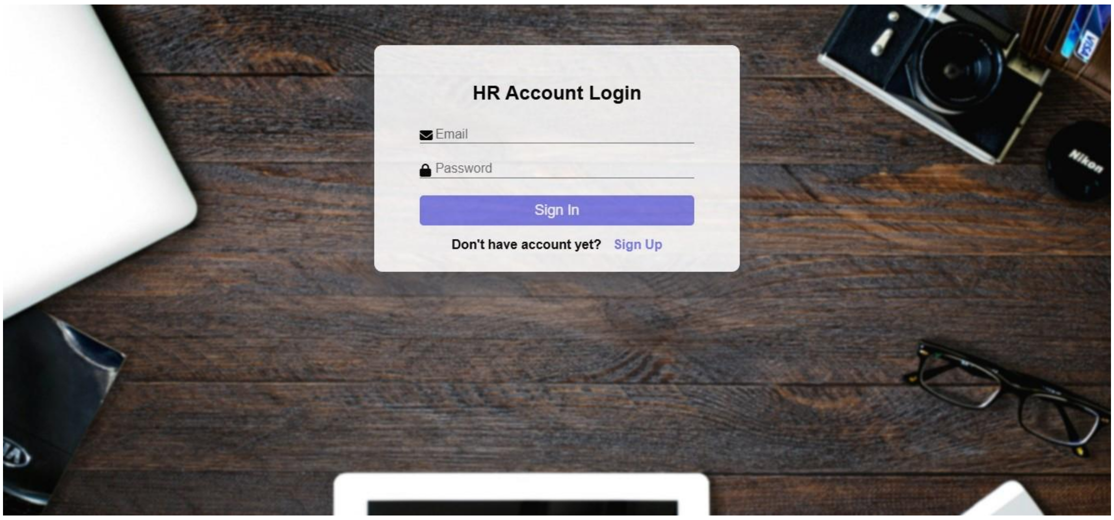

Figure 6.2.1: Login Page

6.2.2 Dashboard

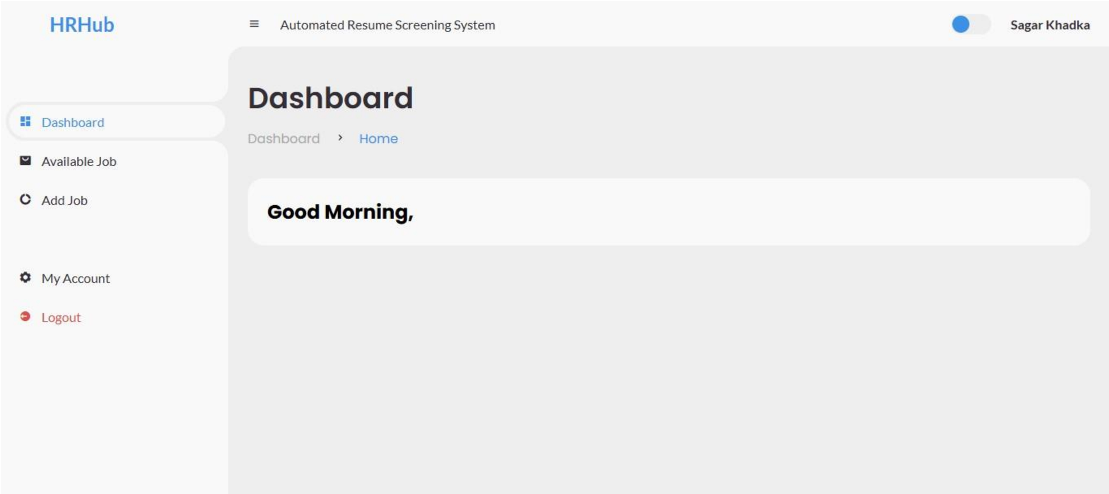

Figure 6.2.2 Dashboard

6.2.3 Final Output

Below is the output generated by our automated resume screening system.

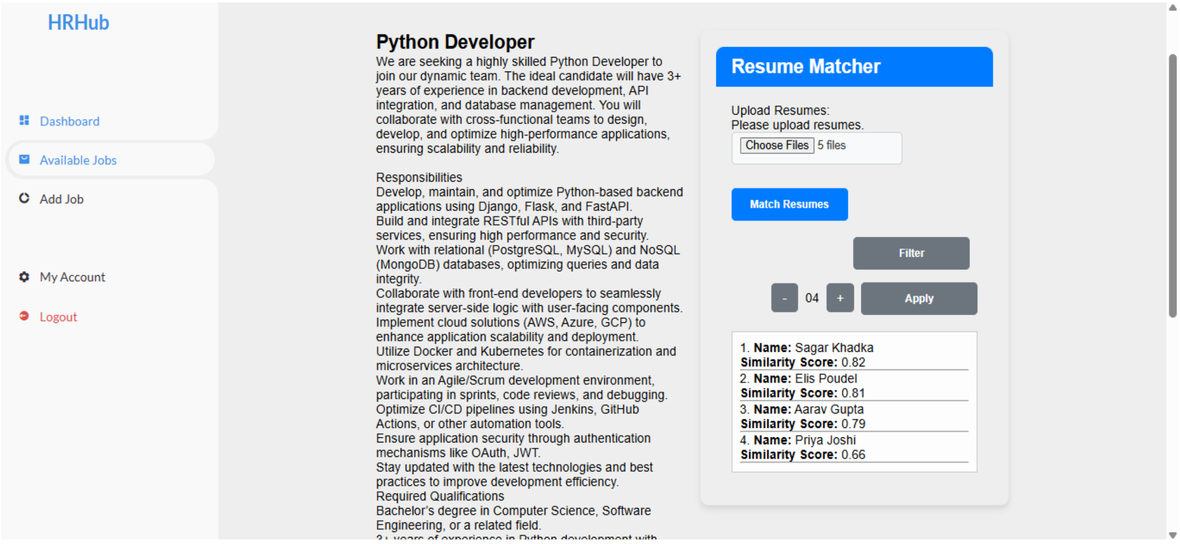

Figure 6.2.3: Final Output

CHAPTER 7

CONCLUSION

The Automated Resume Screening System designed in our project will try to solve some of the challenges in the recruitment process by reducing inefficiency and enhancing transparency. By utilizing advanced technologies such as NLP and machine learning, this system will automate the extraction and matching of relevant candidate information with job requirements, hence speeding up the recruitment process with increased accuracy and fairness for both HR teams. The platform approach in the system empowers an HR team to optimize the talent acquisition process much more easily and reliably.

CHAPTER 8

LIMITATIONS

Below are the limitations of our project:

i. **Data Quality Dependence:**

It requires structured resumes and job descriptions to work; the ones which are not so well-structured will introduce errors.

ii. **Limitations of Format:**

Might not work that well with non-standard formats, such as scanned images or handwritten resumes; additional processing would be required.

iii. **Language barrier:**

It might not all cover languages or dialects; thus, there might be some limitation to global recruitment.

iv. **Data Confidentiality:**

much personal information which requires strict security and compliance is under the auspices of legislation, such as GDPR; not doing so is a source of legal complication.

v. **Limited Context:**

While it reviews for skills and keywords, this application can make no judgments for such subjective qualities as soft skills, cultural fit, or personality traits.

CHAPTER 9

FUTURE ENHANCEMENTS

Utilizing the following methods could significantly enhance the model's performance:

i. Multi-language Support: increased support for a variety of languages to ensure maximum international recruitment.
ii. Integration with the job portal: feeding from LinkedIn, Indeed, and all other similar websites for instant processing of resumes.
iii. Video Resume Analysis: Add video resumes to save time and make more appropriate assessments.
iv. Improved Understanding with NLP: Stronger algorithms for understanding context, sentiment, and soft skills.
v. Mobile Application Development: The development of the mobile application provides an opportunity for job seekers and recruiters to have access to the system anytime.

REFERENCES

1. Shushanta Pudasaini, S. Shakya, Sagar Lamichhane, S. Adhikari, A. Tamang, and S. Adhikari, “Scoring of Resume and Job Description Using Word2vec and Matching Them Using Gale–Shapley Algorithm,” Lecture notes in networks and systems, pp. 705–713, Jul. 2021, doi: https://doi.org/10.1007/978-981-16-2126-0_55.
2. Varun, “Calculating Document Similarities using BERT, word2vec, and other models,” *Medium*, Sep. 26, 2020. :/towardsdatascience.com/calculating-document-similarities-using-bert-and-other-models-b2c1a29c963https0.
3. C. Daryani, G. S. Chhabra, H. Patel, I. K. Chhabra, and R. Patel, “AN AUTOMATED RESUME SCREENING SYSTEM USING NATURAL LANGUAGE PROCESSING AND SIMILARITY,” ETHICS AND INFORMATION TECHNOLOGY, Jan. 2020, doi: https://doi.org/10.26480/etit.02.2020.99.103.
4. K. Dang, “Job — Resume Matching — Part 1 /2: Obtaining similarity score using Doc2Vec,” *Medium*, Jun. 24, 2023. https://medium.com/job-resume-matching-part-1-2-obtaining-similarity-score-using-doc2vec-a6d07fe3b355.
5. R. Iriondo, “Amazon Scraps Secret AI Recruiting Engine that Showed Biases Against Women - Machine Learning - CMU - Carnegie Mellon University,” Machine Learning | Carnegie Mellon University, Oct. 11, 2018. https://www.ml.cmu.edu/news/news-archive/20162020/2018/october/amazon-scraps-secret-artificial-intelligence-recruiting-engine-that-showed-biases-against-women.html.
6. Calculating Document Similarities using BERT, word2vec, and other models https://arxiv.org/pdf/1810.04805.pdf, Accessed: 2024-12-23, 2024.
7. Sidorov, G., Gelbukh, A., Gómez-Adorno, H. and Pinto, D., 2014. Soft similarity and soft cosine measure: Similarity of features in vector space model. Computación y Sistemas, 18(3),491-504.
8. Protean Studios, The agile model and its application in software development, https: //proteanstudios.com/the-agile-model-and-its-applicationin-software-development, Accessed: 2024-12-11, 2024.

APPENDIX : SOURCE CODE

```
# Function to clean extracted text
def clean_text(text):
    """
    Cleans extracted text by:
    - Lowercasing
    - Removing special characters, punctuation, and extra spaces
    - Removing stopwords
    """
    text = text.lower()  # Convert to lowercase
    text = re.sub(r'\s+', ' ', text)  # Remove extra spaces
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)  # Remove special characters
    words = word_tokenize(text)  # Tokenize words
    cleaned_text = ' '.join([word for word in words if word not in stopwords.words('english')])
# Remove stopwords
    return cleaned_text

# Function to extract user name using Named Entity Recognition (NER)
def extract_name(text):
    """
    Extracts the first detected PERSON entity from the resume text.
    """
    doc = nlp(text)
    for ent in doc.ents:
        if ent.label_ == "PERSON":
            return ent.text  # Return the first detected name
    return "Unknown"  # Default if no name is found

# Function to extract text from PDF
def extract_text_from_pdf(file_path):
    text = ""
    with open(file_path, 'rb') as file:
        reader = PyPDF2.PdfReader(file)
```

```
        for page in reader.pages:
            page_text = page.extract_text()
            if page_text:
                text += page_text + " "
    cleaned_text = clean_text(text)
    return cleaned_text, extract_name(text)

# Function to extract text from DOCX
def extract_text_from_docx(file_path):
    text = docx2txt.process(file_path)
    cleaned_text = clean_text(text)
    return cleaned_text, extract_name(text)

# Function to extract text from TXT
def extract_text_from_txt(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        text = file.read()
    cleaned_text = clean_text(text)
    return cleaned_text, extract_name(text)

# General function to extract text and name from different file formats
def extract_text(file_path):
    if file_path.endswith('.pdf'):
        return extract_text_from_pdf(file_path)
    elif file_path.endswith('.docx'):
        return extract_text_from_docx(file_path)
    elif file_path.endswith('.txt'):
        return extract_text_from_txt(file_path)
    else:
        return "", "Unknown"

# Function to match resumes with job description using SBERT + TF-IDF.
# Now accepts an optional job_title parameter.
```

```
def match_resumes(job_description, resume_files, job_title=None):
    resumes = []
    names = []

    # If a job title is provided and it's not already present in the job description,
    # prepend it with the label "Job Title:".
    if job_title and "job title" not in job_description.lower():
        job_description = f"Job Title: {job_title}\n{job_description}"

    for resume in resume_files:
        text, name = extract_text(resume)
        resumes.append(text)
        names.append(name)

    if not resumes or not job_description:
        return [], "Please upload resumes and enter a job description."

    # Clean job description before processing
    job_description = clean_text(job_description)

    # TF-IDF Similarity
    vectorizer = TfidfVectorizer().fit_transform([job_description] + resumes)
    vectors = vectorizer.toarray()
    job_vector = vectors[0]
    resume_vectors = vectors[1:]
    tfidf_similarities = cosine_similarity([job_vector], resume_vectors)[0]

    # SBERT Similarity
    job_embedding = sbert_model.encode(job_description, convert_to_tensor=True)
    resume_embeddings = sbert_model.encode(resumes, convert_to_tensor=True)
    sbert_similarities = np.array(util.pytorch_cos_sim(job_embedding,
resume_embeddings).cpu().numpy())[0]
```

```
    # Combined Score (80% SBERT + 20% TF-IDF)
    combined_scores = (sbert_similarities * 0.8) + (tfidf_similarities * 0.2)

    # Sort all resumes by score in descending order
    top_indices = combined_scores.argsort()[::-1]
    top_resumes = [(resume_files[i], names[i], round(combined_scores[i], 2)) for i in
top_indices]

    return top_resumes, "Top matching resumes:"
```