

RESUME PARSER USING NLP

SDP PROJECT REPORT

*Submitted in partial fulfillment for the
award of the degree of*

Bachelor of Technology
In
Computer Science & Engineering
by

K.Mahalakshmi(21BCE9653)

B.K.Geetha(21BCE9665)

M.V.Lakshmi Swathi (21BCE8582)

Guided by

Prof.Bommareddy Lokesh

Professor SCOPE, VIT-AP



AMARAVATI

SCOPE

MAY, 2025

DECLARATION

I hereby certify that the thesis titled "**Resume Parser Using NLP**" submitted by me in partial fulfillment for the award of the Bachelor's degree is the result of original work carried out under the guidance of **Prof. Bommareddy Lokesh**.

I also affirm that the contents of this thesis have not been submitted, in whole or in part, for the award of any other degree or diploma at this institute or any other university/institution.

Place: VIT-AP UNIVERSITY

Date: 08/05/2025

TEAM MEMBERS

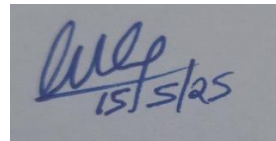
K.Mahalakshmi-21BCE9653

B.K.Geetha-21BCE9665

M.V.Lakshmi Swathi-21BCE8582

CERTIFICATE

This is to certify that the thesis titled “**Resume Parser Using NLP**”, submitted by **K. Mahalakshmi (21BCE9653), B.K. Geetha (21BCE9665), and M.V. Lakshmi Swathi (21BCE8582)**, is submitted in partial fulfillment of the requirements for the award of the **Bachelor of Technology** degree. This work is a genuine record of the research carried out under my supervision. The contents of this thesis, whether in full or in part, have neither been copied from any external source nor submitted to any other institute or university for the award of any degree or diploma.

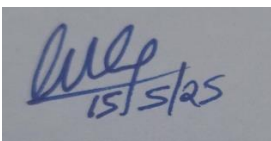


Dr. Bommareddy Lokesh

Guide

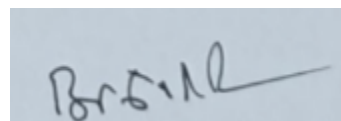
✓

This thesis is satisfactory/unsatisfactory



Internal Examiner 1

Dr. Bommareddy Lokesh



Internal Examiner 2

Dr. Naga Jagadesh Bommagani

Approved by

HoD, Department of School of
Computer Science and Engineering

Dr. Reeja SR

ABSTRACT

With the advancement of online recruiting systems, job application websites have made it easier than ever for candidates to upload their resumes with just a few clicks. While this convenience benefits job seekers, it has also led to a massive influx of resume submissions. As a result, human resource departments are now facing significant challenges in managing the high volume of applications, screening candidates efficiently, and identifying the most suitable individuals for specific roles. Adding to this complexity Resumes are presented in various formats, incorporating diverse writing styles, fonts, sizes, colors, layouts, and file types. This inconsistency makes it difficult for recruiters to read and evaluate every resume thoroughly.

To address these challenges, I propose the development of a **resume parser system using Natural Language Processing (NLP)**. This system will assist HR professionals and It involves the automatic extraction of essential information from resumes, including personal details, skills, educational background, work history, certifications, and notable accomplishments. By structuring and standardizing resume data, this tool can help streamline the shortlisting process, minimize human error, and significantly reduce the time and effort required for manual resume screening. Ultimately, it will improve the overall efficiency and accuracy of the hiring process.

In addition, this system can be enhanced with machine learning models to rank candidates based on job relevance, flag missing or misleading information, and even match candidates to the most suitable roles within an organization.

ACKNOWLEDGEMENT

It is with great pleasure and sincere gratitude that I express my heartfelt thanks to **Prof. Bommareddy Lokesh**, Professor at **SCOPE, VIT-AP**, for his constant guidance, unwavering support, and valuable insights throughout this project. More than anything, his patience and encouragement have been truly inspiring. My association with him has extended beyond academics—working under his mentorship has been a remarkable opportunity to learn from an expert and intellectual in the field of SCOPE. I would like to express my gratitude to Dr. G. Viswanathan, Mr. Sankar Viswanathan, Dr. S. V. Kota Reddy, and Dr. CH. Pradeep Reddy, SCOPE, For providing a supportive environment to work in and for his constant inspiration throughout the duration of the course.

With a joyful heart, I sincerely thank all the teaching staff and members of our university, whose dedicated efforts and timely encouragement greatly contributed to my academic journey. Their unwavering support and enthusiasm played a vital role in helping me acquire the necessary knowledge to successfully complete my course. I am also deeply grateful to my parents for their constant support and motivation throughout this endeavor.

I am truly grateful to my friends for their encouragement and motivation, which inspired me to take up and complete this project. Lastly, I extend my heartfelt thanks to everyone who supported me, directly or indirectly, in the successful completion of this work.

TEAM MEMBERS

K.MAHALAKSHMI-21BCE9653

B.K.GEETHA-21BCE9665

M.V.LAKSHMI SWATHI-21BCE8582

TABLE OF CONTENTS

Sl. NO.	Chapter Title	Page No.
1	ABSTRACT	4
2	ACKNOWLEDGEMENT	5
3	LIST OF FIGURES AND TABLES	6-8
4	1 INTRODUCTION 1.1 Background 1.2 Problem Survey 1.3 Goals and Objectives	9 12 16 18
5	2 LITERATURE SURVEY 2.1 NLP in Recruitment 2.2 Resume Parsing Techniques 2.3 Challenges and Solutions	20 24 26 28
6	3 PROPOSED METHODOLOGY 3.1 Dataset Collection and Annotation 3.2 Training the NLP Model 3.3 Web Development	30 30 31 32
7	4 PROJECT FLOW 4.1 Performance Requirements 4.2 Feasibility Report	37 37 38
8	5 SYSTEM REQUIREMENTS 5.1 Purpose of the System 5.2 Problems in the Existing System 5.3 Solution of these Problems	41 41 41 42
9	6 RESULTS AND EVALUATION 6.1 Performance Metrics 6.2 Case Studies 6.3 User FeedBack	48 48 48 49
10	7 FUTURE SCOPE	51
11	8 CONCLUSION 8.1 Summary of Findings 8.2 Challenges Faced	52 52 53
12	9 APPENDICIES	54
13	10 REFERENCES	58

LIST OF FIGURES

Sl. No.	Chapter Title	Page No.
1	Fig 1 – Work Flow	15
2	Fig 2 - Methodology	25
3	Fig 3 – Login Page	33
4	Fig 4 – Home Page	34
5	Fig 5 – Job Posting	34
6	Fig 6 - Application Form	35
7	Fig 7 – Resume uploader	36
8	Fig 8 – Score (Resume Parser)	36
9	Fig 9 – Env Setup	48
10	Fig 10 – Source Codes	49

LIST OF ACRONYMS

Sl. No.	Acronym
1	NLP-NATURAL LANGUAGE PROCESSING
2	HR-HUMAN RESOURCE
3	NER-NAMED ENTITY RECOGNITION
4	UI-USER INTERFACE
5	NLTK-NATURAL LANGUAGE TOOLKIT
6	REGEX-REGULAR EXPRESS

CHAPTER-1

INTRODUCTION

The advancement of online recruiting systems has revolutionized the way candidates apply for jobs, allowing them to effortlessly upload their resumes to various job application websites. This convenience, however, has led to a significant increase in the number of resumes submitted, presenting a major challenge for human resource (HR) departments. HR professionals now face the daunting task of reviewing an overwhelming volume of resumes, each varying in format, writing style, font choice, and other stylistic elements. This variation makes it difficult to efficiently and accurately assess each candidate's qualifications.

To address this issue, we propose the development of a resume parser utilizing natural language processing (NLP) techniques. This system aims to assist HR departments and recruiters by extracting the essential information from resumes, streamlining the applicant review process, and reducing the likelihood of errors. By automating the extraction of crucial information—such as work experience, educational background, skills, and contact details—the resume parser facilitates HR professionals to focus on selecting the best candidates for job positions more effectively and efficiently. This solution not only enhances the recruitment process but also ensures a more consistent and fair evaluation of all applicants.

In the current competitive job environment, employers often receive a large number of resumes for each job opening. Manually reviewing these resumes to identify the ideal candidate is both time-intensive and susceptible to mistakes. Each resume is unique, with candidates using different formats, fonts, and layouts to present their qualifications. This diversity can obscure the critical

information HR professionals need to make informed decisions, leading to inefficiencies and potential biases in the hiring process.

Traditional methods of resume screening, such as keyword searches or basic filters, often fall short in capturing the nuanced details that differentiate a qualified candidate from an unqualified one. Additionally, the large number of resumes can overwhelm HR teams, slowing down the hiring process and increasing the risk of overlooking qualified candidates. There is a clear need for a more sophisticated, automated solution that can handle the complexities of resume parsing and provide HR professionals with the information they need in a structured, easily accessible format.

Natural Language Processing (NLP) is a field of artificial intelligence that deals with enabling computers to understand and interact with human language. By leveraging NLP techniques, a resume parser can be developed to automatically extract and organize information from resumes, regardless of their format. This parser is capable of detecting and organizing important resume sections like personal details, professional experience, educational background, skills, and certifications.

The proposed resume parser would work by first preprocessing the resume to standardize its format. This involves converting different file types (e.g., PDF, DOCX) into a readable text format. The parser would then use NLP algorithms to analyze the text, identifying key phrases and patterns that correspond to different sections of the resume. For example, it could recognize job titles, company names, dates of employment, and educational institutions. By extracting this information, the parser can create a structured profile for each candidate that HR professionals can easily review and compare.

Automating the resume screening process significantly reduces the time required to review each application. HR professionals can concentrate on

assessing the most qualified candidates instead of sorting through numerous resumes. By standardizing the information extracted from resumes, the parser minimizes the risk of human error and ensures that no critical details are overlooked. This leads to more accurate assessments of each candidate's qualifications. By using consistent criteria for every resume, the resume parser guarantees a fair and impartial evaluation process. This consistency helps to eliminate any potential biases that may arise from manual resume screening. The parser is designed to process a high volume of resumes, making it ideal for organizations ranging from small startups to large corporations. As the number of applications grows, the parser can scale accordingly to maintain its performance. By streamlining the recruitment process, companies can respond to applicants more quickly, improving the overall candidate experience. This responsiveness can enhance the company's reputation and attract more high-quality candidates.

Implementing a resume parser with NLP involves several steps, including data collection, model training, and system integration. The first step is to gather a diverse set of resumes to train the NLP model. This dataset should include resumes from different industries, job levels, and formats to ensure the parser can handle a wide range of inputs. Next, the NLP model needs to be trained to recognize and extract relevant information from the resumes. This involves annotating the training data with labels for different sections and using machine learning algorithms to learn the patterns and features associated with each section. Once the model is trained, it can be integrated into a larger system that includes preprocessing tools, a user interface for HR professionals, and a database to store the parsed information.

While the potential benefits of a resume parser are significant, there are also challenges to consider. One of the main challenges is ensuring the parser can handle the variability in resume formats and styles. This requires a robust NLP

model that can generalize well across different inputs. Additionally, the parser must be able to accurately interpret the context of the information it extracts, such as distinguishing between job titles and company names or identifying the correct dates of employment. Another challenge is maintaining the privacy and security of the candidates' information. The resume parser must comply with data protection regulations and implement appropriate measures to safeguard the data it processes.

The development of a resume parser using natural language processing offers a promising solution to the challenges faced by HR departments in the modern recruitment landscape. By automating the extraction of key information from resumes, the parser can improve the efficiency, accuracy, and fairness of the hiring process. This not only benefits HR professionals but also enhances the candidate experience, helping companies attract and retain top talent. As the demand for effective recruitment solutions continues to grow, the resume parser stands out as a valuable tool for the future of HR management.

1.1 BACKGROUND

With the rapid advancement of online recruiting systems, job application processes have become more streamlined, allowing candidates to easily upload their resumes to job application websites. This convenience, however, has led to a significant increase in the volume of resumes submitted, posing substantial challenges for human resource (HR) departments. The sheer number of resumes to review can overwhelm HR professionals, making it difficult to efficiently and accurately identify the most suitable candidates.

Moreover, resumes submitted by candidates come in various formats, including different writing styles, fonts, sizes, and colors. This variability adds an

additional layer of complexity to the review process, as HR departments must navigate through inconsistent formatting to extract relevant information.

Given these challenges, traditional manual review methods are often inefficient and prone to errors. The task of reading and evaluating each resume in its entirety is not only time-consuming but also susceptible to human error, which can lead to missed opportunities for identifying qualified candidates.

To address these issues, there is a growing need for automated solutions that can enhance the efficiency and accuracy of the recruitment process. One promising approach is the use of Natural Language Processing (NLP) for resume parsing. NLP technologies can analyze and interpret the textual content of resumes, enabling the automated extraction of key information such as qualifications, skills, and experiences.

The proposed resume parser project leverages NLP to assist HR departments in overcoming these challenges. By automating the extraction of detailed information from resumes, the parser aims to streamline the recruitment process, reduce manual effort, and minimize errors. The application includes functionalities such as resume upload, keyword extraction, and job matching, all designed to improve the efficiency of candidate evaluation and enhance the overall recruitment workflow.

In conclusion, incorporating NLP into resume parsing marks a major breakthrough in recruitment technology, providing an effective way to address the challenges brought by the growing number and diversity of resumes during hiring.

1.1.1 OBJECTIVE

The objective of this project is to develop an automated resume parser using Natural Language Processing (NLP) to improve the efficiency and accuracy of the recruitment process. The primary goals include:

- 1.** Streamlining Resume Review: Automate the extraction of key information from resumes to reduce the time and effort required for manual review by HR professionals.
- 2.** Enhancing Candidate Evaluation: Improve the accuracy of identifying suitable candidates by systematically analyzing and interpreting resume content.
- 3.** Reducing Errors: Minimize human errors in the recruitment process by providing a consistent and objective analysis of resumes.
- 4.** Facilitating Job Matching: Enable efficient matching of candidates to job openings based on extracted keywords and qualifications, ensuring better alignment between job requirements and candidate profiles.

1.1.2 OVERVIEW

This project focuses on leveraging Natural Language Processing (NLP) to develop a sophisticated resume parsing application. NLP is a field of artificial intelligence focused on enabling computers to comprehend, interpret, and produce human language in a meaningful manner. Extract Key Information: Analyze the textual content of resumes to identify and extract relevant details such as skills, qualifications, experiences, and achievements.

- Standardize Data Transform resumes from multiple formats into a standardized structure that simplifies processing and analysis.

- **Match Candidates to Jobs:** Use the extracted information to match candidates with appropriate job openings based on predefined criteria.
- **Resume Upload:** Allowing candidates to submit their resumes in various formats through an easy-to-use interface.
- **Keyword Extraction:** Identifying and extracting important keywords and phrases from the resumes that are crucial for evaluating qualifications and skills.
- **Job Matching:** Using the extracted data to compare against job descriptions and identify the best fit for each position.

By implementing this resume parser, HR departments will benefit from a more efficient and accurate recruitment process, ultimately leading to better hiring decisions and reduced administrative burden.

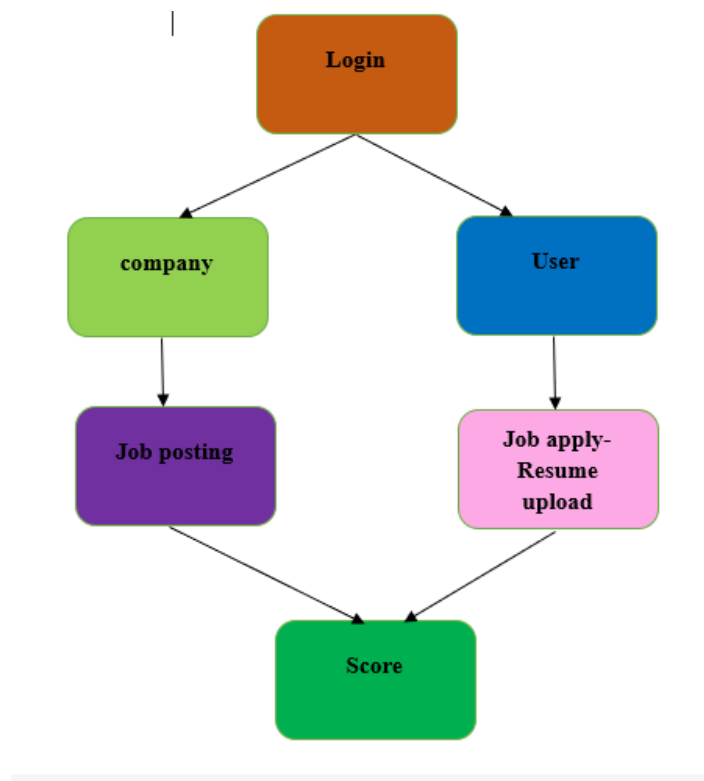


FIG:WORK FLOW

1.2 PROBLEM

1.2.1 CHALLENGES

Recruiters and human resource departments face significant difficulties when processing large volumes of resumes manually. These challenges include:

1. **Time Consumption:** Reviewing each resume individually is time-consuming, especially when dealing with hundreds or thousands of submissions. This extensive manual effort delays the recruitment process and increases the time to hire.
2. **Inconsistent Formatting:** Resumes are often submitted in various formats with diverse writing styles, fonts, sizes, and colors. This inconsistency makes it difficult for recruiters to standardize and compare resumes effectively.
3. **Information Overload:** Resumes can be lengthy and filled with a vast amount of information. Identifying key qualifications, skills, and experiences amidst this overload requires considerable effort and can lead to information being overlooked.
4. **Human Error:** Manual resume reviews can lead to mistakes caused by fatigue, unconscious bias, or oversight, resulting in recruiters potentially overlooking key information or misjudging candidates, which may affect hiring outcomes.
5. **Scalability Issues:** As the number of job applications grows, manually managing and reviewing resumes becomes increasingly unmanageable. This scalability issue further strains the recruitment process and impacts overall efficiency.

1.2.2 NEED

To address these challenges, there is a pressing need for automated solutions that can streamline the recruitment process. Automated resume parsing using Natural Language Processing (NLP) offers a promising solution by:

1. **Enhancing Efficiency:** Automating the extraction of key information from resumes significantly reduces the time and effort required for manual review. This enables recruiters to dedicate more time to important activities like interviewing and assessing the most promising candidates.
2. **Standardizing Data:** NLP techniques can convert resumes into a standardized format, making it easier to compare and analyze candidate qualifications across different submissions.
3. **Improving Accuracy:** Automated parsing minimizes human error and bias by providing a consistent and objective analysis of resume content. This leads to more accurate identification of qualified candidates.
4. **Scaling with Volume:** Automated solutions can handle large volumes of resumes efficiently, allowing recruiters to manage growing application pools without compromising on the quality of candidate evaluation.

In summary, adopting an automated resume parsing solution is essential for overcoming the challenges of manual resume processing and achieving a more efficient, accurate, and scalable recruitment process.

1.3 GOALS AND OBJECTIVES

GOALS

1. **Enhance Recruitment Efficiency:** Streamline the resume review process to significantly reduce the time and effort required for manual evaluation, enabling HR departments to handle larger volumes of resumes more effectively.
2. **Improve Job Matching Accuracy:** Develop an automated system that accurately matches candidates with job openings based on their skills, qualifications, and experiences, leading to better alignment between job requirements and candidate profiles.
3. **Reduce Errors:** Minimize human errors and biases in the recruitment process by providing a consistent, objective analysis of resumes.
4. **Facilitate Scalable Recruitment:** Create a solution that scales with the growing number of job applications, ensuring that the recruitment process remains efficient even as application volumes increase.

OBJECTIVES

1. **Develop a Resume Parser:** Create a robust resume parsing tool capable of accurately extracting key information from resumes, such as qualifications, skills, experiences, and achievements.
2. **Integrate NLP Techniques:** Utilize Natural Language Processing (NLP) techniques, including Named Entity Recognition (NER), keyword extraction, and text classification, to analyze and interpret resume content effectively.

3. **Design a User-Friendly Interface:** Build an intuitive and user-friendly interface that allows candidates to easily upload their resumes and HR professionals to interact with parsed data and job matching results.
4. **Implement Job Matching Algorithm:** Develop and integrate an algorithm that matches extracted candidate information with job descriptions, facilitating the identification of suitable candidates for each job opening.
5. **Ensure Data Security and Privacy:** Implement robust security measures to protect candidate data and ensure compliance with data privacy regulations.
6. **Test and Validate the System:** Conduct thorough testing to validate the accuracy and performance of the resume parser, including testing with diverse resume formats and job descriptions to ensure reliability and effectiveness.

By achieving these goals and objectives, the project aims to create a powerful tool that enhances the efficiency and accuracy of the recruitment process, benefiting both HR departments and job seekers.

CHAPTER- 2

LITERATURE SURVEY

Kinge, Bhushan, et al proposed a system that analyzes resumes, extracting skills and certifications, and scraping data from LinkedIn and GitHub profiles to predict suitable job roles and offer improvement recommendations. The Indian recruitment market has grown significantly due to increased job openings and demand for cost-effective labor, leading to the rise of specialized recruitment companies. Manual resume screening remains time-consuming, prompting the use of machine learning models to rank resumes based on job relevance. The accuracy of various resume recommendation methods ranges from 78.53% to 98.96%, highlighting the potential of automated systems to enhance efficiency and support both recruiters and job seekers.[1]

Sruthi Patlolla and colleagues developed an intelligent resume screening system that leverages Natural Language Processing (NLP) to extract key information from unstructured resumes. The system identifies relevant skills and qualifications to recommend appropriate job titles. It also offers personalized suggestions to help improve resumes, enhancing their appeal to potential employers. Built with Python and various libraries, the system saves the extracted data and assigns ratings based on its analysis, providing targeted feedback for improvement. The process starts by accepting resumes in PDF or Word formats and utilizes the pyresparser module, which depends on spacy and NLTK for data extraction. This innovative method aims to transform the job application experience by making it more streamlined and effective for both applicants and recruiters.

use of machine learning models to rank resumes based on job relevance. The accuracy of various resume recommendation methods ranges from 78.53% to 98.96%, highlighting the potential of automated systems to enhance efficiency and support both recruiters and job seekers.[1]

Suresh and Yeresime proposed a system automates resume fetching, categorization, and extraction of critical information from unstructured PDF resumes. The job market's growth and increased applications have made job recommendation and candidate selection complex. It utilizes machine learning techniques, including logistic regression and Gaussian Naïve Bayes, to provide job recommendations and suggest improvements to resumes. The system's performance is evaluated on classification accuracy and recommendation effectiveness, representing a significant advancement in automating these processes.[3]

Mankawade, V. Pungliya, R. Bhonsle, S. Pate, A. Purohit and A. Raut proposed a system that develops a job recommender system using NLP and machine learning to automate job searching. It extracts information from PDF resumes, trains models with logistic regression and Gaussian Naïve Bayes, and scrapes job listings from Naukri.com. Jobs are ranked based on cosine similarity between user skills and job requirements. The system shows high accuracy in predicting suitable professions and recommending jobs, streamlining the job search process.[4]

Mittal, Vrinda, et al proposed a system that explores the use of Natural Language Processing (NLP) and machine learning for automating resume parsing and candidate evaluation. It involves extracting structured data from unstructured resumes in various formats, followed by preprocessing steps like tokenization, stop-word removal, and stemming or lemmatization. The study

uses algorithms like SVM and XGBoost, with XGBoost showing superior performance. Key tasks include extracting personal information, skills, and qualifications from resumes, and applying machine learning models for categorization and recommendations. The approach improves efficiency and accuracy in the hiring process, reducing manual effort and bias.[5]

Gunawardana, Stephan proposed system develops a four-tier web application that parses user resumes to extract keywords and uses these keywords to find job opportunities via an API. Users can upload resumes, and the system matches job listings based on extracted skills. The application includes user authentication with sign-up and sign-in, storing user profiles and preferences securely in a modern backend database. It employs Vue for the frontend and Firebase/Firestore/Google Storage for the backend, reflecting significant learning and experience in web development.[6]

Resume Analyzer Using Text Processing – This literature review introduces an efficient Company Recommender System that leverages text mining and machine learning techniques to assist recruiters in identifying the most suitable candidates for a given role. When applicants upload their resumes, the system evaluates and ranks them based on the specific requirements of the company. These rankings help organizations streamline the selection process by focusing on top-matching candidates. [7]

Resume parsing is a technology designed to retrieve essential details from resumes and convert them into a structured format. It works by scanning documents to identify key information such as contact details, educational background, professional experience, and skill sets. Once extracted, this data is categorized into predefined sections, making it easier to store, search, and analyze. By automating the information extraction process across diverse resume formats, resume parsing significantly improves the efficiency of

recruitment, allowing recruiters to handle and evaluate applications more effectively. [8]

The article provides techniques for extracting email addresses, phone numbers, and links from text using regular expressions (regex). Email addresses are identified with patterns that match common formats, such as `[\w\.-]+@[\w\.-]+\.\w+`. Phone numbers are detected using patterns that accommodate various formats, including international codes and separators, exemplified by `(\+?\d{1,4}[\s.-]?)?(?(\d{1,4})?[\s.-]?(\d{1,4}[\s.-])?\d{1,9})`. Links are extracted with patterns for URLs that start with `http://`, `https://`, or `www.`, followed by domain names and optional paths, like `https?:\/\/[^\s]+`. These regex patterns facilitate the efficient retrieval of contact details and hyperlinks from textual data.[9]

Bhor, Shubham, et al The proposed model aims to enhance the hiring process by automating resume parsing and ranking using Natural Language Processing (NLP). It involves building a job portal where users can upload resumes, which are then processed to extract and structure relevant information based on three modules: employee, company, and admin. It uses Optical Character Recognition (OCR) to convert resumes into a standard text format, and NLP techniques—including lexical, syntactic, and semantic analysis, and Named Entity Recognition (NER)—to parse and interpret the data. The parsed resumes are scored and ranked based on the job requirements defined by companies. Additionally, the system integrates data from social media platforms like LinkedIn and utilizes Elasticsearch for data management and query-based sorting, providing a dashboard for HR to view and manage candidate information efficiently.[10]

2.1 NLP in RECRUITMENT

OVERVIEW

- The use of Natural Language Processing (NLP) in recruitment has attracted considerable interest for its ability to revolutionize the way resumes are reviewed and analyzed. Studies and existing research emphasize various critical areas where NLP has brought meaningful advancements:
- **Automated Resume Screening:** NLP techniques have been employed to automate the initial screening of resumes, reducing the manual effort required and improving the efficiency of the recruitment process.
- **Candidate Matching:** NLP tools have been used to match candidate profiles with job descriptions, helping recruiters identify the best-fit candidates based on skills, experiences, and qualifications.
- **Text Classification and Sentiment Analysis:** Natural Language Processing (NLP) has significantly advanced the capabilities of recruitment systems by enabling sophisticated text classification and sentiment analysis techniques. Through these methods, recruiters can gain a deeper and more nuanced understanding of both candidate profiles and job descriptions. Text classification helps in automatically categorizing resumes based on job roles, skills, or qualifications, thereby reducing manual effort and enhancing accuracy. Sentiment analysis, on the other hand, can be used to interpret the tone and intent behind a candidate's language, offering additional insights into personality traits, motivation, and communication style. These technologies collectively contribute to a more informed and efficient decision-making in hiring.

TECHNIQUES

Several NLP techniques are utilized in parsing resumes:

- **Named Entity Recognition (NER):** NER helps detect and categorize important elements like names, companies, dates, and places within resumes. This technique helps extract important information from unstructured text.
- **Keyword Extraction:** This technique involves identifying and extracting relevant keywords and phrases from resumes that match job requirements.
- **Part-of-Speech Tagging:** Part-of-speech tagging helps in understanding the grammatical structure of resumes, which aids in extracting and classifying various sections such as skills, qualifications, and experiences.
- **Text Classification:** Text classification assigns predefined categories to resume content, such as education, work experience, and skills, making it easier to analyze and compare resumes.
- **Entity Linking:** Entity linking connects identified entities to a knowledge base, enriching the extracted information with additional context.

In addition to these, techniques like semantic similarity analysis are used to compare the content of resumes with job descriptions based on contextual meaning rather than just keyword matches. Dependency parsing helps to understand the relationships between words, which is useful for capturing detailed experience or role descriptions. Topic modeling techniques can also be applied to identify major areas of expertise in a candidate's profile. These advanced methods enable a deeper understanding of resume content, enhancing the accuracy and efficiency of automated resume screening systems.

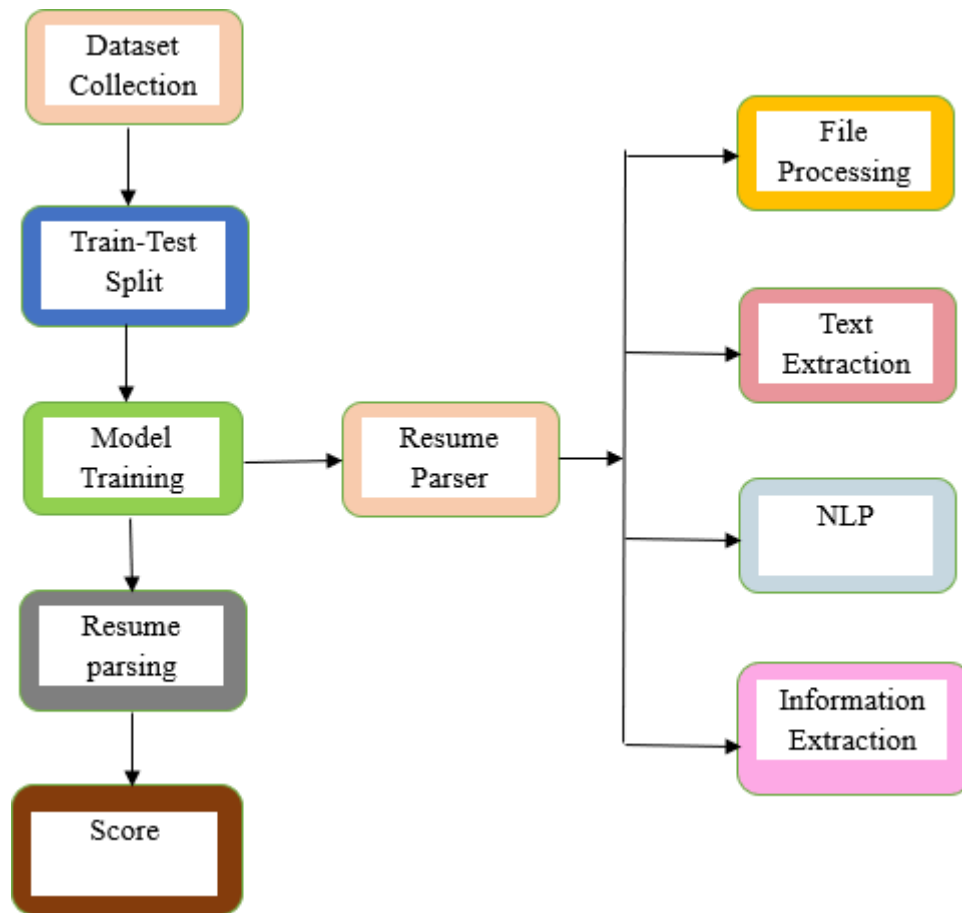


FIG 2: Methodology

2.2 RESUME PARSING TECHNIQUES

TRADITIONAL METHODS

Traditional resume parsing methods involve:

- **Rule-Based Systems:** Early resume parsers used rule-based systems that rely on predefined patterns and keywords to extract information. Although these systems are simple to use, they struggle with flexibility and often fail to manage different resume formats in efficient manner in proper way and optimized.

- **Keyword Matching:** Keyword matching involves searching for specific keywords within resumes. This method can be effective but often fails to account for context and synonyms, leading to incomplete or inaccurate extractions.

MODERN NLP APPROACHES

Modern NLP approaches have advanced the capabilities of resume parsing:

- **Machine Learning Models:** Machine learning models, such as Support Vector Machines (SVM) and Random Forests, have been used to improve the accuracy of resume parsing by learning from labeled data and adapting to different resume formats.
- **Deep Learning Techniques:** Techniques such as Recurrent Neural Networks (RNNs) and Transformers (e.g., BERT) have enhanced the ability to understand and process complex resume content, enabling more accurate extraction of information.
- **Contextual Analysis:** Modern NLP approaches consider the context in which keywords and entities appear, improving the precision of information extraction and understanding of resume content.

ADVANCEMENTS

Recent advancements in NLP that have improved resume parsing capabilities include:

- **Contextual Embeddings:** The use of contextual embeddings, such as those generated by BERT, allows for a deeper understanding of the nuances in resume text, leading to more accurate parsing.

- **Transfer Learning:** Transfer learning techniques enable NLP models to leverage pre-trained knowledge on large corpora and adapt it to specific resume parsing tasks, enhancing performance and reducing the need for extensive training data.
- **Multi-Task Learning:** Multi-task learning approaches enable models to perform multiple related tasks simultaneously, such as extracting skills and classifying job roles, improving overall parsing accuracy.

2.3 CHALLENGERS and SOLUTIONS

COMMON ISSUES

Resume parsing faces several challenges:

- **Varied Formats:** Resumes come in diverse formats, including different fonts, styles, and structures, making it difficult for parsers to standardize and interpret information.
- **Unstructured Data:** Resumes often contain unstructured data, such as free-form text, which can be challenging to analyze and extract meaningful information from.
- **Synonym and Context Variability:** The use of synonyms and varying contexts in resumes can complicate keyword matching and information extraction.

PROPOSED SOLUTIONS

To address these challenges, several solutions and improvements have been proposed:

- **Adaptive Parsing Models:** Developing adaptive parsing models that can handle varied formats and structures by learning from diverse resume datasets.

- **Advanced NLP Techniques:** Utilizing advanced NLP techniques such as contextual embeddings and deep learning to improve the understanding of unstructured data and variability in resumes.
- **Hybrid Approaches:** Combining rule-based and machine learning approaches to leverage the strengths of both methods and enhance parsing accuracy.
- **Continuous Learning:** Implementing continuous learning strategies that allow the parser to evolve and improve over time based on new resume data and feedback.

These advancements and solutions aim to overcome the limitations of traditional resume parsing methods and enhance the effectiveness of NLP in the recruitment industry.

These advancements and solutions aim to overcome the limitations of traditional resume parsing methods and enhance the effectiveness of NLP in the recruitment industry. In the long run, these systems can be integrated into end-to-end recruitment platforms to not only parse resumes but also match candidates to roles, track applicant progress, and assist in decision-making. Additionally, incorporating explainable AI techniques can help recruiters understand why certain candidates were selected or ranked higher, increasing transparency and trust in automated systems.

CHAPTER-3

PROPOSED METHODOLOGY

3.1 DATASET COLLECTION AND ANNOTATION

DATASET COLLECTION:

- Sources: Gather resumes from various sources, including online job boards, company databases, and publicly available datasets. Ensure diversity in formats (PDF, Word, plain text) and styles (chronological, functional, combination).
- Volume: Aim for a large and varied dataset to improve model generalization. A minimum of several thousand resumes is recommended.

ANNOTATION:

- Annotation Process: Resumes are manually labeled to highlight essential details like contact information, skills, professional experience, education certifications, and other important sections
- Tools: Use annotation tools like Prodigy, or custom-built solutions to streamline the annotation process and ensure consistency.
- Guidelines: Develop detailed annotation guidelines to ensure uniformity across annotators. Include definitions and examples for each annotation category.
- Annotator Training: Train annotators thoroughly on guidelines and tools to minimize errors and ensure high-quality annotations.

QUALITY ASSURANCE:

- Inter-Annotator Agreement: Measure inter-annotator agreement using metrics like Cohen's Kappa to ensure consistency and reliability of annotations.
- Review Process: Implement a review process where senior annotators or subject matter experts validate and correct annotations.

- **Sample Audits:** Regularly audit random samples of annotated data to check for accuracy and adherence to guidelines.
- **Feedback Loop:** Create a feedback loop where annotators can discuss ambiguous cases and refine guidelines based on real-world examples.

3.2 TRAINING THE NLP Model

MAINTENANCE AND CONTINUOUS IMPROVEMENT:

- **Regular Updates:** Continuously update the model with new data to keep it relevant and improve performance.
- **User Feedback:** Collect and analyze feedback from HR personnel and job seekers to identify pain points and areas for enhancement.
- **Iteration:** Iteratively refine the model and system based on feedback, error analysis, and advancements in NLP research.

3.3 WEB DEVELOPMENT

To develop an efficient recruitment system, the methodology involves gathering requirements from HR personnel, job seekers, and IT staff to identify the functional and non-functional needs. We then design an architecture, database schema, and APIs, followed by creating a responsive and intuitive UI using technologies like React.js. The backend, built with Spring Boot and Thymeleaf integrates with an NLP-powered resume parser using tools like NLTK and SpaCy to extract and structure candidate information. We collect and annotate a diverse resume dataset, train and evaluate the NLP model, and deploy it on a scalable cloud service. The system includes robust authentication mechanisms to secure user data. Regular feedback and continuous improvement cycles incorporate user insights, enhancing the system with advanced natural language processing and customizations for specific roles or industries.

Data Collection

Due to the advancement of online recruiting systems, candidates can easily upload their resumes on job application websites. This results in a large number of resumes being submitted, posing a significant challenge for human resource departments to review and select the best candidates.

For this project, resumes were collected in various formats such as PDF, DOCX, and plain text from online sources and job portals.

Preprocessing

Resumes are presented in multiple formats, featuring diverse writing styles, font types, sizes, and color schemes. To handle this diversity, the following preprocessing steps were applied:

1. Text Extraction: Using tools like Apache Tika and PyPDF2, text was extracted from different resume formats.
2. Normalization: Converting all text to a uniform format (e.g., lowercasing, removing special characters).
3. Tokenization: Breaking down the text into separate words or tokens.
4. Stop Words Removal: Removing common words that do not contribute to the resume's key information (e.g., "and", "the").
5. Lemmatization/Stemming: Reducing words to their base or root form to ensure consistency.

NLP Techniques Applied

To assist human resource departments or recruiters in extracting detailed information from resumes, several NLP techniques were employed:

- **Named Entity Recognition (NER):** Detecting and categorizing important entities such as names, dates, skills, and job titles with tools like spaCy and NLTK.
- **Part-of-Speech (POS) Tagging:** Identifying the grammatical parts of speech (nouns, verbs, adjectives) in the text to understand the structure and meaning of sentences.
- **Keyword Extraction:** Using algorithms like TF-IDF (Term Frequency-Inverse Document Frequency) and RAKE (Rapid Automatic Keyword Extraction) to identify the most relevant keywords and phrases.
- **Clustering:** Grouping similar resumes based on extracted features to facilitate easier comparison and ranking of candidates.

Tools and Technologies Used

The project utilized a range of tools and technologies to develop and deploy the resume parser:

- **Frontend:** Vue.js was used to create an interactive and user-friendly interface for uploading resumes and displaying results.
- **Backend:** Firebase/Firestore provided a scalable and secure backend for storing user data. Google Storage was used for handling document uploads.
- **NLP Libraries:** SpaCy and NLTK were the primary libraries used for implementing NLP techniques such as NER, POS tagging, and keyword extraction.
- **Additional Tools:** Apache Tika and PyPDF2 were used for text extraction from different document formats, and Python's Pandas library facilitated data manipulation and preprocessing.

By using these methodologies, the project aims to streamline the resume screening process, reduce errors, and assist human resource departments in efficiently identifying the best candidates for job positions.

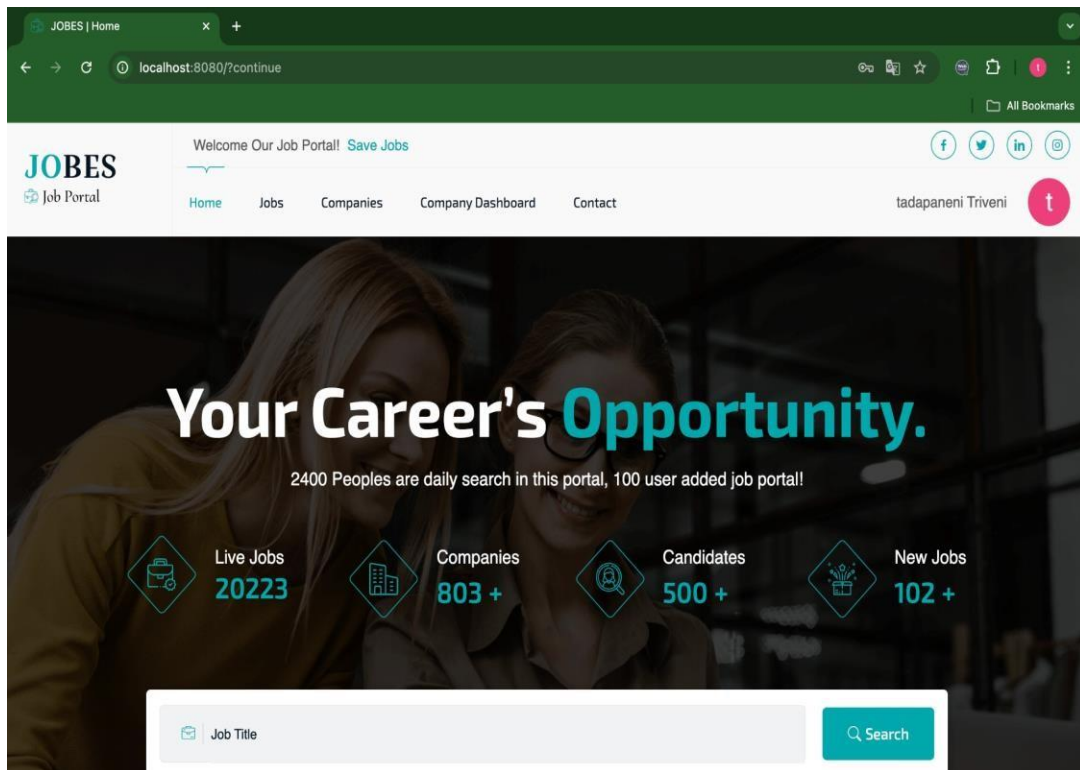


FIG.3: Home page

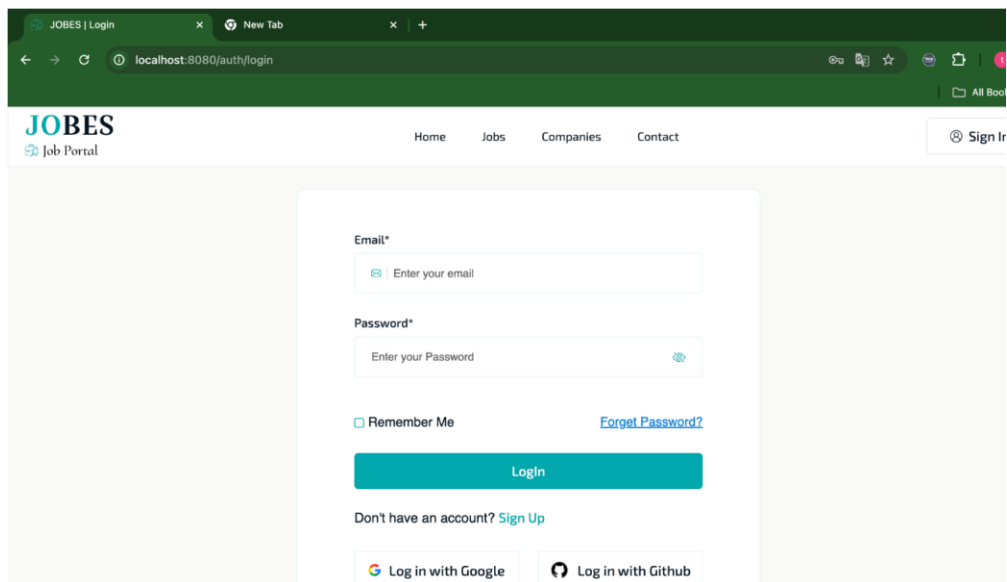


FIG.4: Login form

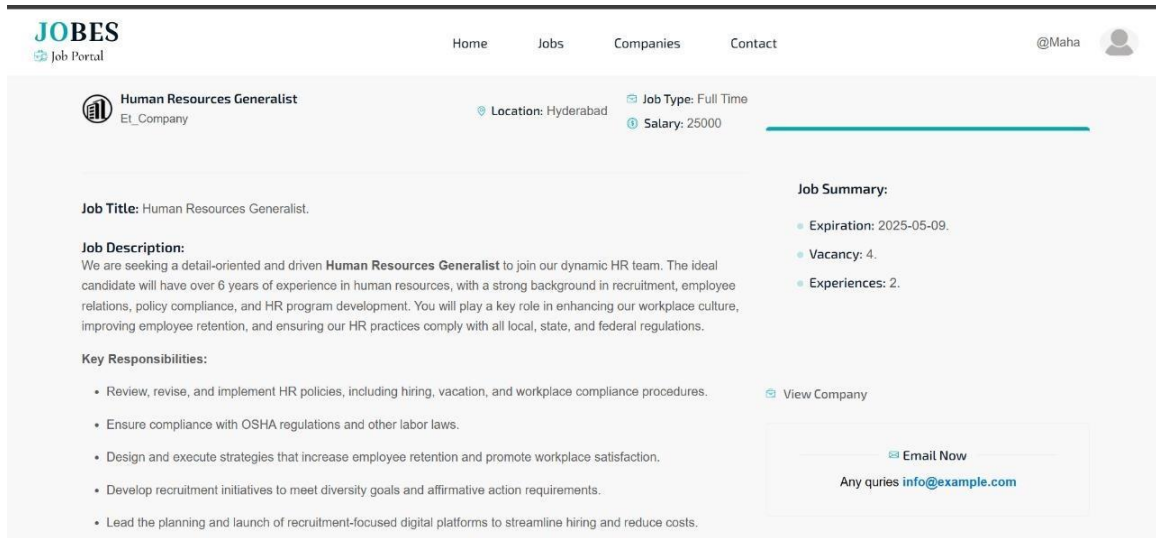


FIG. 5: Job Posting

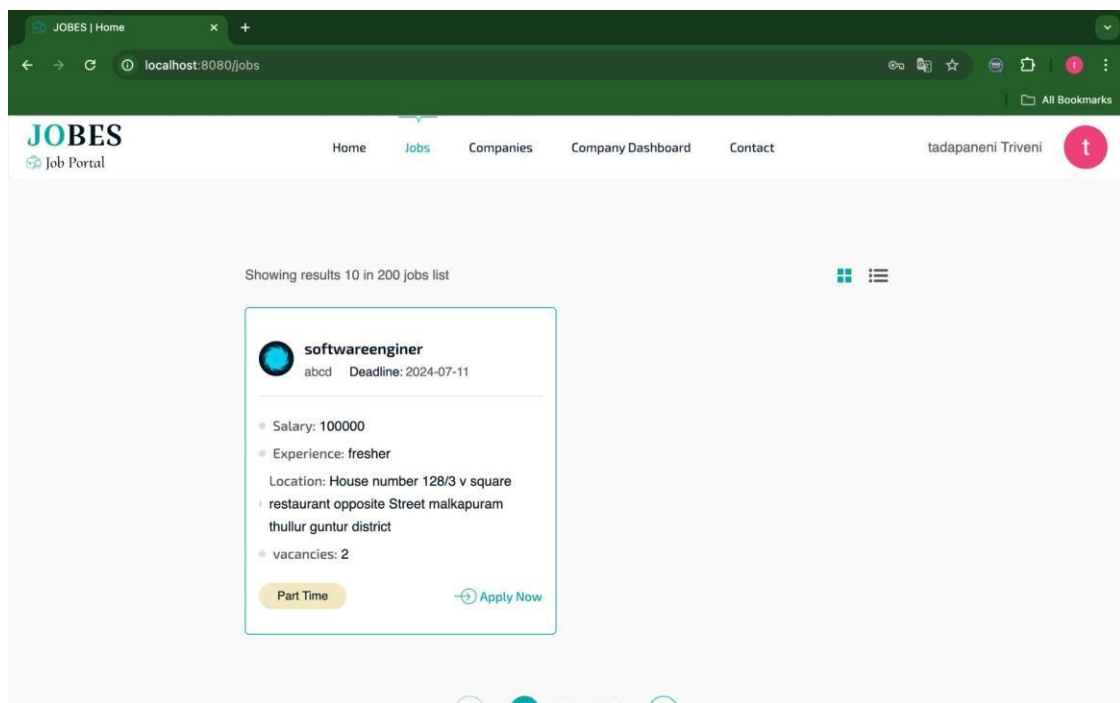


FIG. 6: Apply form

JOBES
Job Portal

Home Jobs Companies Contact

tadapaneni Triveni

Apply Here..!

Job Title*

full stack developer

Resume*

Choose file No file chosen

Apply

FIG 7 : Resume upload form

Resume Parser Controller Send the job description and resume to get similarity score.

POST /resume-parser/get-similarity

Parameters

Try it out

Name	Description
resume file (formData)	Choose File No file chosen
job_description string (formData)	job_description

Responses

Response content type: application/json

Code	Description
200	Success

POST /resume-parser/parse

FIG 8: Score

CHAPTER-4

PROJECT FLOW

4.1 PERFORMANCE REQUIREMENTS:

The performance requirements for the Resume Parser system using NLP are designed to ensure efficiency, accuracy, scalability, user experience, integration, security, reliability, and continuous improvement. The system must handle and process resumes in various formats quickly, with upload and parsing taking only a few seconds, even when multiple users are active simultaneously. Accuracy is paramount, with the document parsing engine expected to extract key details like contact information, skills, work experience, and educational background with over 90% precision, and the Named Entity Recognition (NER) component achieving at least 85% precision and recall rates. Scalability is crucial, as the system must manage large volumes of resumes without performance degradation, and the database must support efficient data management and quick retrieval.

Efficiency: Quick resume processing within a few seconds, handling multiple concurrent users. Accuracy: Over 90% precision in extracting key details; NER component achieving at least 85% precision and recall. Scalability: Capable of managing large volumes of resumes with consistent performance; efficient database management. User Experience: Intuitive and responsive UI, easy navigation, interactive visualizations, and a user feedback mechanism.

Integration: Compatibility with existing HR systems through APIs and integration points; customizable parsing algorithms for specific roles or industries. Security: Robust data protection, secure login, password policies, data encryption, and access control for authorized users only.

Reliability: High system uptime, effective error handling with clear messages and logging.

Continuous Improvement: Machine learning integration for ongoing accuracy and efficiency improvements, monitored by key performance metrics:

4.2 FEASIBILITY REPORT:

Preliminary Investigation:

Examining Project Feasibility

The objective of the preliminary investigation is to assess the feasibility of the Resume Parser using NLP project, determining its potential usefulness to the organization. Technical Feasibility: The technical feasibility examines whether the proposed system can be implemented using existing technology and resources. Key points include:

The technical feasibility: It evaluates if the proposed system can be developed with the current technology and resources. Important aspects include: Technology Availability – The required tools, such as NLP libraries like NLTK and SpaCy, are accessible to build the resume parser.

Equipment Capacity: The proposed system can handle the required data, with hardware requirements including an Intel Pentium processor, 512 MB RAM, and 20GB hard disk. System Response: The system is designed to provide adequate response times, regardless of the number or location of users, Upgradeability: The system can be upgraded as needed to incorporate new features or improve performance.

Accuracy, Reliability, Security: The system ensures accuracy, reliability, ease of access, and data security. The use of robust security measures, such as secure login mechanisms and session management, guarantees these aspects. The current system is web-based, providing easy access to users, and utilizes open-source software and existing equipment, making it technically feasible.

Operational Feasibility

Operational feasibility evaluates whether the proposed system will meet the organization's operating requirements and be accepted by users. Key points include:

Management and User Support: There is sufficient support from both management and users for the system. System Usage: The system is expected to be used effectively once developed and implemented, with minimal resistance from users. User Requirements: User requirements and management issues have been considered to ensure acceptance and optimal performance. The well-planned design of the system ensures optimal utilization of resources and improves performance

Economic Feasibility

Economic feasibility assesses whether the system is a good financial investment for the organization. Key points include: Cost-Benefit Analysis: The development costs are evaluated against the ultimate benefits derived from the system. Financial benefits must equal or exceed the costs Existing Resources: The system utilizes existing hardware and software, requiring no additional expenditures.

Nominal Expenditure: Developing the user interface with currently available technologies helps keep the costs low, which makes the project economically viable. The minimal financial outlay required for development, combined with the substantial advantages the system offers, demonstrates that this investment is practical and beneficial for the organization. By leveraging existing tools and resources, the project avoids unnecessary expenses while promising significant returns in efficiency and effectiveness, confirming its sound economic feasibility.

CHAPTER-5

SYSTEM REQUIREMENTS

5.1 PURPOSE OF THE SYSTEM

The purpose of this project is to develop a resume parser using natural language processing to assist the human resource department or recruiters in efficiently extracting detailed information from resumes. This tool aims to streamline the applicant review process by standardizing the diverse formats in which resumes are submitted, thereby reducing the workload and minimizing errors in selecting the best candidates for job positions.

5.2 PROBLEMS IN THE EXISTING SYSTEM:

- 1. High Volume of Resumes:** With the ease of online resume submissions, HR departments receive an overwhelming number of resumes, making it difficult to efficiently review and process each one.
- 2. Diverse Resume Formats:** Applicants provide resumes in diverse formats, featuring a variety of writing styles, fonts, sizes, colors, and file types. This diversity complicates the review process as it requires HR personnel to spend extra time standardizing and interpreting the information.
- 3 Manual Review Challenges:** The manual review of resumes is time-consuming prone to human error. HR staff must read through each resume in detail to extract relevant information, increasing the likelihood of missing critical details or misinterpreting the data.

3. Inefficient Candidate Selection: The inconsistency in resume formats and the manual extraction of information make it challenging to compare candidates objectively and efficiently. This inefficiency can delay the selection process and potentially result in overlooking qualified candidates.

4. Integration Issues with HR Systems: Existing systems may lack seamless integration capabilities with HR platforms, making it difficult to transfer parsed resume data into applicant tracking systems and other HR tools.

5. Lack of Feedback Mechanism: Current systems often lack a robust feedback mechanism for users to report issues or suggest improvements in the resume parsing process. This absence of feedback can hinder the continuous improvement of the system's accuracy and efficiency.

6. Security Concerns: Ensuring the security and confidentiality of sensitive personal and professional information shared by job seekers is critical. Existing systems may not have adequate security measures in place to protect against unauthorized access and data breaches.

5.3 SOLUTION OF THESE PROBLEMS:

1. Automated Resume Parsing: Develop a resume parser using Natural Language Processing (NLP) to automatically retrieve essential details from resumes. This reduces the manual effort required by HR personnel, allowing them to handle a higher volume of resumes efficiently.

2. Efficient Information Extraction: The resume parser is capable of extracting organized data like contact information, skills, work history, and education from unstructured text. This automation minimizes the risk of human error and ensures that all relevant information is captured accurately.

3. Objective Candidate Comparison: By presenting parsed data in a standardized and structured format, the system enables objective comparison of candidates. HR personnel can quickly identify the most qualified candidates based on consistent criteria.

4. Seamless Integration with HR Systems: Develop APIs or integration points to ensure compatibility with existing HR platforms. This facilitates the smooth transfer of parsed resume data into applicant tracking systems and other HR tools, streamlining the overall recruitment process.

5. Feedback and Improvement Mechanism: Include a feedback feature for users to report issues or suggest improvements in the parsing process. This could be a simple form or a dedicated section within the application. Regularly updating the parsing algorithms based on user feedback will enhance accuracy and efficiency over time.

6. Robust Security Measures: Implement strong security protocols to protect sensitive information. Secure login mechanisms, password policies, and session management should be enforced to prevent unauthorized access and ensure the confidentiality and integrity of personal data shared by job seekers and HR personnel.

SCOPE OF THE PROJECT:

The project aims to develop an NLP-based resume parser to streamline the recruitment process by automatically extracting key information from resumes in various formats. It will feature advanced NLP techniques for entity recognition, skill extraction, and semantic analysis, ensuring consistent and

accurate data extraction. The system will include a user- friendly interface for resume uploads, seamless integration with existing HR systems, and robust security measures to protect sensitive information. Additionally, it will offer customization for specific job roles, incorporate user feedback for continuous improvement, and be designed for scalability and high performance to handle a large number of resumes efficiently.

FUNCTIONAL COMPONENTS OF THE PROJECT:

Resume Upload Feature: A user-friendly interface that allows candidates and recruiters to upload resumes in various formats, including PDFs. This component should handle different file types and sizes efficiently.

Document Parsing Engine: Utilize NLP libraries such as NLTK and spaCy to parse the content of the uploaded resumes. This system must efficiently extract organized information such as contact details, skills, job experience, and educational qualifications from unstructured text.

Named Entity Recognition (NER): Implement advanced NLP techniques to identify and classify entities within the resume text. This includes recognizing names, titles, companies, locations, dates, and other relevant information that helps in understanding the candidate's profile.

Skill Extraction and Tagging: Develop algorithms to identify and tag skills mentioned in the resume. This could involve keyword extraction and semantic analysis to understand the context in which skills are mentioned.

Integration with HR Systems: Ensure compatibility with existing HR systems by providing APIs or integration points. This will facilitate seamless data exchange between the resume parser and HR platforms.

Feedback and Improvement Mechanism: Include a feature for users to provide feedback on the parsing accuracy and suggest improvements. This could involve a simple form or a dedicated feedback section within the application.

Customization and Machine Learning Integration: Allow customization of parsing algorithms for specific job roles or industries. Additionally, integrate machine learning algorithms to continuously improve the accuracy and efficiency of the parsing process over time.

THE MODULES INVOLVED ARE:

Human Resources:-

The Human Resources module serves as the backbone of the recruitment process, managing the flow of job vacancies, candidate applications, and the selection process. It leverages the resume parser to streamline the initial screening phase, reducing the workload on HR staff by automating the extraction of essential candidate information from resumes. This module ensures that HR personnel can focus on strategic aspects of recruitment, such as interviewing and talent development, by providing them with a structured overview of each candidate's qualifications and experiences. Additionally, it facilitates the integration of the resume parser with existing HR systems, allowing for seamless data management and reporting.

Job seekers:-

The Job Seekers module caters to the needs of individuals seeking employment opportunities. It provides a platform for job seekers to upload their resumes in various formats, ensuring compatibility with the resume parser's capabilities.

This module emphasizes ease of use, allowing candidates to quickly and efficiently submit their applications. By integrating with the resume parser, it enables job seekers to have their resumes processed and analyzed for relevance to posted jobs, enhancing their chances of being considered for positions that match their skills and experience. The module also includes features to guide job seekers through the application process, providing feedback on submission status and next steps.

Resume parser:-

The Resume Parser module utilizes advanced Natural Language Processing (NLP) techniques to extract and analyze information from resumes uploaded by job seekers. It is designed to handle a wide range of resume formats and styles, ensuring comprehensive data extraction regardless of the input medium. Key features include the ability to parse contact details, skills, work experience, and educational backgrounds, leveraging technologies such as NLTK and SpaCy for deep analysis. The module outputs parsed data in a structured format, facilitating easy integration with HR systems and databases. Future enhancements may include machine learning integration for enhanced analysis and categorization, as well as customization features tailored to specific job roles or industries.

Web app:-

The User Interface (UI) module focuses on creating an intuitive and user-friendly interface for both HR personnel and job seekers. It ensures that the process of uploading resumes, accessing parsed information, and navigating through job postings is straightforward and accessible. The UI incorporates responsive design principles to accommodate users across various devices, enhancing the overall user experience. It also includes features for displaying

parsed resume data in an easily digestible format, such as interactive visualizations, to aid HR personnel in evaluating candidate profiles effectively.

Authentication:-

The Authentication module is crucial for securing the application and ensuring that only authorized users can access sensitive information and functionalities. It implements robust security measures, including secure login mechanisms, password policies, and session management, to protect user data and prevent unauthorized access. This module is integral to maintaining trust within the system, as it guarantees the confidentiality and integrity of personal and professional information shared by job seekers and HR personnel alike.

CHAPTER-6

RESULTS AND EVALUATION

6.1 PERFORMANCE METRICS

To evaluate the effectiveness of the resume parser, several performance metrics were assessed:

1. Accuracy:

- **Precision and Recall:** Measured the precision (proportion of true positive results among all positive results) and recall (proportion of true positive results among all actual positive cases) of the Named Entity Recognition (NER) and keyword extraction components
- **F1 Score:** Calculated the F1 score, the harmonic mean of precision and recall, to gauge overall performance

2. Efficiency:

- **Processing Time:** Evaluated the average time taken to parse a resume and extract relevant information.
- **Scalability:** Tested the system's ability to handle large volumes of resumes simultaneously without significant degradation in performance.

3. Error Rate:

- **False Positives/Negatives:** Analyzed the rate of incorrect extractions or mismatches to identify areas for improvement.

6.2 CASE STUDIES

Examples of successful parsing and matching demonstrated the effectiveness of the system:

1. Case Study 1: Tech Industry Recruitment

- **Resume:** A software engineer with extensive experience in Java and python.
- **Outcome:** The parser accurately extracted key skills, work experience, and education. The candidate was correctly matched to several relevant job opportunities in tech companies, leading to successful placements

2. Case Study 2: Healthcare Sector

- **Resume:** A healthcare administrator with a focus on project management and regulatory compliance.
- **Outcome:** The parser identified specialized skills and certifications. The candidate was matched with several job postings requiring similar expertise, resulting in multiple interview requests.

3. Case Study 3: Finance Sector

- **Resume:** A financial analyst with expertise in data analysis and financial modelling
- **Outcome:** The parser effectively extracted relevant experience and qualifications. The candidate was successfully matched to roles requiring similar skills, highlighting the parser's ability to handle diverse resumes.

6.3 USER FEEDBACK

Insights from end users revealed the following:

1. Positive Feedback:

- **Efficiency:** Users appreciated the reduced time and effort required to screen resumes manually.
- **Accuracy:** Many HR professionals noted the high accuracy of extracted information and its relevance to job requirements

- **User Interface:** The intuitive design of the Vue.js-based interface was praised for its ease of use.

2. Areas for Improvement:

- **Resume Formatting:** Some users reported issues with resumes that had complex formatting, suggesting a need for improved handling of such formats.
- **Customization:** Users asked for additional customization features to better tailor the parsing and matching criteria according to specific job roles and industry requirements.
- **Integration:** There were suggestions for integrating the parser with existing Applicant Tracking Systems (ATS) to streamline workflows further.

Overall, the results indicate that the resume parser is effective in improving the recruitment process, though there are areas where further refinement could enhance its capabilities and user satisfaction.

CHAPTER-7

FUTURE SCOPE

Multilingual Support: Expanding the parser to support multiple languages will make it more versatile and applicable in global recruitment scenarios.

Support for Diverse Formats: As resume formats evolve, future resume parsers will be able to handle a wider variety of formats, including multimedia resumes and interactive portfolios, ensuring no applicant is overlooked due to format incompatibility.

Incorporation of Psychometric Testing

Behavioral Insights: By integrating psychometric testing into the resume screening process, Recruiters can obtain a more comprehensive evaluation of candidates by understanding their personality traits, cognitive skills, and cultural compatibility, going beyond simply assessing skills and experience.

Expanded Data Sources

Social Media Integration: Incorporate information from professional social media platforms like LinkedIn to enhance candidate profiles and offer a more detailed overview of their qualifications.

Job Board Integration: Connect with multiple job boards to provide candidates with a broader range of job opportunities based on the parsed resume data.

CHAPTER-8

CONCLUSION

With the advancement of online recruitment, employers receive a large volume of resumes. As a result, reviewing and hiring suitable candidates has become a significant challenge for HR departments. To address this, an automated intelligent system based on Natural Language Processing (NLP) has been developed. This system can convert resumes in various formats into text and successfully extract key information. Additionally, it can compare the applicant's resume with the job description to calculate a similarity score. This tool helps HR teams or employers efficiently screen resumes before interviews, aiding in the selection of the best candidates for job openings.

8.1 SUMMARY OF FINDINGS

The development of an NLP-based resume parser aimed to enhance the efficiency and accuracy of the recruitment process. The project successfully achieved the following outcomes:

- **Efficient Data Extraction:** The parser successfully retrieved important information from resumes, such as personal details, skills, professional experience, and educational background.
- **Improved Candidate Matching:** By leveraging keyword extraction and clustering techniques, the system accurately matched candidates to job opportunities based on their qualifications and experience.

- **User-Friendly Interface:** The use of HTML(Thymeleaf) for the frontend resulted in an intuitive and interactive user interface, simplifying the resume upload and review process for HR departments.
- **Scalable Backend:** Springboot provided a reliable and scalable backend solution, ensuring secure storage and easy access to user data and resume information.

8.2 CHALLENGES FACED

During the development of the resume parser, several challenges were encountered:

- **Diverse Resume Formats:** Handling the variety of resume formats, including different file types, writing styles, and formatting, required robust preprocessing techniques to ensure consistent data extraction.
- **Named Entity Recognition (NER) Accuracy:** Ensuring high accuracy in NER was challenging due to the variations in how information is presented in resumes. Fine-tuning the NER models to improve precision was necessary.
- **Handling Ambiguities:** Resolving ambiguities in text, such as differentiating between job titles and skills, required sophisticated NLP techniques and additional contextual analysis.
- **Scalability and Performance:** Processing large volumes of resumes efficiently while maintaining performance was a critical challenge. Optimization of algorithms and database queries was essential to achieve scalability.

User Data Security: Ensuring the security and privacy of user data, especially sensitive personal information contained in resumes, required implementing robust security measures and compliance with data protection regulations. Despite these challenges, the project successfully delivered a functional and efficient resume parser that significantly aids the recruitment process by reducing manual effort and improving candidate selection accuracy.

CHAPTER -9

APPENDICES

Source Code:

5.1- Environment Setup and Library Installation

```
[ ] # https://spacy.io/usage/
!pip install spacy_transformers
!pip install -U spacy

!pip install pdfminer-six
```

To build the resume parser, essential libraries like SpaCy, spacy_transformers, pdfminer-six, and NumPy were installed. spaCy was used for Named Entity Recognition (NER), and Spacy_transformers allowed the use of transformer models like BERT. pdfminer-six was employed to extract text from PDF resumes.

There were some compatibility issues with the versions of libraries, so updates and downgrades were done. NumPy was set to version 1.23.5, transformers to 4.36.2, and SpaCy to 3.7.2. These changes ensured smooth functionality and resolved any errors.

5.2- Converts Annotated Data to SpaCy Doc Format

```
def get_spacy_doc(data):
    nlp = spacy.blank('en')
    db = DocBin()
    for text, annot in tqdm(data):
        doc = nlp.make_doc(text)
        ents = []
        entity_indices = []
        for start, end, label in annot['entities']:
            if any(i in entity_indices for i in range(start, end)):
                continue
            entity_indices += list(range(start, end))
            span = doc.char_span(start, end, label=label, alignment_mode="strict")
            if span:
                ents.append(span)
        doc.ents = ents
        db.add(doc)
    return db
```

The `get_spacy_doc(data)` function is used to convert raw training data into a format that SpaCy can understand. It takes each text and its labeled entities, creates a document object (doc), and then adds the named entity annotations to it.

This processed data is saved using SpaCy's DocBin, which stores it in a compact format. This is later used to train the Named Entity Recognition (NER) model effectively.

5.3- Train a SpaCy Named Entity Recognition (NER) model

```
!python -m spacy \
  train /content/drive/MyDrive/resume-parser/data/config.cfg \
  --output /content/drive/MyDrive/resume-parser/output \
  --paths.train /content/drive/MyDrive/resume-parser/data/train.spacy \
  --paths.dev /content/drive/MyDrive/resume-parser/data/test.spacy \
  --gpu-id 0
```

This command trains a SpaCy model for Named Entity Recognition (NER). It uses the configuration file (config.cfg) to define settings like the pipeline and hyperparameters. The training data, stored in train.spacy, helps the model learn to identify named entities such as names, emails, and phone numbers.

Validation data (test.spacy) is used to evaluate the model during training. The trained model is saved in the output directory. The `--gpu-id 0` argument ensures the training uses the GPU, speeding up the process.

5.3- Extracts Phone Number

```
def extract_contact_number_from_resume(text):
    pattern = r"\b(?:\+?\d{1,3}[-.\s]?\\d{3})?(?:[-.\s]?\\d{3}[-.\s]?\\d{4})\b"
    match = re.search(pattern, text)
    return match.group() if match else None
```

This function is designed to extract a candidate's contact number directly from the resume text. It uses a regular expression (regex) pattern that matches most common phone number formats, including formats with dashes, spaces, or country codes.

Once a match is found, the function returns the phone number. This is especially useful in automation, where contact details need to be pulled from resumes without manual checking. If no number is found, the function returns None.

5.4- Extracts Email ID

```
def extract_email_from_resume(text):  
    pattern = r"\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b"  
    match = re.search(pattern, text)  
    return match.group() if match else None
```

This function helps in automatically extracting an email address from the resume using a regex pattern. It searches for valid email formats, such as any characters followed by an @ and a domain (like .com, .org, etc.). The regex is designed to handle common email structures.

When a match is found, the email address is returned to the caller. Email extraction is an essential part of resume parsing, making it easier to save contact details for job communication. This helps automate the process of handling resumes without manual extraction.

5.5- TF-IDF Based Similarity Matching

```
def tfidf_matcher(job_description, resumes):  
    vectorizer = TfidfVectorizer().fit_transform([job_description, resumes])  
    vectors = vectorizer.toarray()  
    similarities = cosine_similarity([vectors[0]], vectors[1:])[0]  
    return similarities
```


This function compares a resume with a job description using TF-IDF (Term Frequency-Inverse Document Frequency), which helps to identify the most important words in each document. Both texts are vectorized into numerical form based on word importance, where common words get less weight and rare but relevant words get more.

It then uses cosine similarity to compute how close the resume is to the job description. A higher similarity score means a better match. This function is useful for recommending the most relevant resumes to recruiters based on job needs.

5.6- Count Vector Matching

```
from sklearn.feature_extraction.text import CountVectorizer

def count_matcher(job_description, resumes):
    vectorizer = CountVectorizer().fit_transform([job_description, resumes])
    vectors = vectorizer.toarray()
    similarities = cosine_similarity([vectors[0]], vectors[1:])[0]
    return similarities
```

This function is a simpler version of text comparison that uses raw word counts instead of TF-IDF weights. It converts the job description and resume into frequency vectors using CountVectorizer, counting how many times each word appears.

Next, cosine similarity is used to measure the degree of similarity between the two documents. While not as precise as TF-IDF, this method still gives a good basic comparison between job requirements and resume content, especially in straightforward text.

REFERENCES

- [1]Kinge, Bhushan, et al. "Resume Screening Using Machine Learning and NLP: A Proposed System." (2022).
- [2]Sruthi, Patlolla, et al. "Smart Resume Analyser: A Case Study using RNN-based Keyword Extraction." E3S Web of Conferences. Vol. 430. EDP Sciences, 2023.
- [3]Suresh, Yeresime. "Machine Learning-Based Recommendations and Classification System for Unstructured Resume Documents." Revue d'Intelligence Artificielle 37.3 (2023).
- [4]A. Mankawade, V. Pungliya, R. Bhonsle, S. Pate, A. Purohit and A. Raut, "Resume Analysis and Job Recommendation," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-5, doi: 10.1109/I2CT57861.2023.10126171
- [5]Mittal, Vrinda, et al. "Methodology for resume parsing and job domain prediction." Journal of Statistics and Management Systems 23.7 (2020): 1265-1274.
- [6] Gunawardana, Stephan. "Resume Parser and Job Search." (2022).
- [7] Literature Reviews - Resume Analyzer Using TextProcessing:
Retrieved from, <https://jespublication.com/upload/2020-110557.pdf>
- [8] What is resume parsing : Retrieved from
<https://www.smartrecruiters.com/resources/glossary/resumeparsing/>

[9] How to extract email address, phone number and links from text: Retrieved from,

<https://zapier.com/blog/extract-links-email-phoneregex/>

[10] Bhor, Shubham, et al. "Resume parser using natural language processing techniques." Int. J.

Res. Eng. Sci 9.6 (2021).

[11] Narendra, G. O., and S. Hashwanth. "Named entity recognition based resume parser and summarizer." Int. J. Adv. Res. Sci. Commun. Technol 2 (2022): 728-735.

[12] Alamelu, M., et al. "Resume validation and filtration using natural language processing." 2021 10th International conference on internet of everything, microwave engineering, communication and networks (IEMECON). IEEE, 2021.

[13] Satheesh, K., et al. "Resume ranking based on job description using SpaCy NER model." International Research Journal of Engineering and Technology 7.05 (2020): 74-77

.