VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

MINI PROJECT-2 REPORT on

RESUME PARSER

Submitted by

JEEVANTHI KASHYAP (1BM21CS080)
JYOTHIKA C N (1BM21CS083)
PALLE PADMAVATHI(1BM21CS125)

Under the Guidance of

Prof. Namratha M
Assistant Professor, BMSCE

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

B. M. S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

March 2024 to June 2024

B. M. S. College of Engineering

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work entitled **"RESUME PARSER"** carried out by **JEEVANTHI KASHYAP (1BM21CS080), J Y O T H I K A C . N (1BM21CS083), P A L L E P A D M A V A T H I (1BM21CS125)** who are bonafide students of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visveswaraiah Technological University, Belgaum during the year 2023-2024. The project report has been approved as it satisfies the academic requirements in respect of **Mini Project-2 (22CS6PWMP2)** work prescribed for the said degree.

Signature of the Guide
Prof. Namratha M
Assistant Professor
BMSCE, Bengaluru

Signature of the HOD
Dr. Jyothi S Nayak
Professor & Head, Dept. of CSE
BMSCE, Bengaluru

External Viva

Name of the Examiner — Signature with date

1._____ _____

2. _____ _____

B. M. S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECALARATION

We, JEEVANTHI KASHYAP (1BM21CS080), JYOTHIKA C.N (1BM21CS083), PALLE PADMAVATHI (1BM21CS080), students of 5th Semester, B.E, Department of Computer Science and Engineering, B. M. S. College of Engineering, Bangalore, here by declare that, this Project Work-1 entitled "Resume Parser" has been carried out by us under the guidance of Prof. Namratha M, Assistant Professor, Department of CSE, B. M. S. College of Engineering, Bangalore during the academic semester March-June 2024.

We also declare that to the best of our knowledge and belief, the development reported here is not from part of any other report by any other students.

Signature

JEEVANTHI KASHYAP (1BM21CS080)

JYOTHIKA C N (1BM21CS083)

PALLE PADMAVATHI(1BM21CS125)

TABLE OF CONTENTS

1. Introduction

In the realm of modern recruitment, the sheer volume of resumes inundating hiring managers presents a formidable challenge, often resulting in overwhelmed teams, delayed decision-making, and potentially overlooked qualified candidates. Traditional resume parsing methods, relying on manual screening or simplistic keyword matching, struggle to efficiently extract pertinent information from these documents. This inefficiency not only prolongs the hiring process but also risks missing out on top talent.

To address this pressing issue, our project endeavors to develop an intelligent resume parsing framework that leverages cutting-edge advancements in Natural Language Processing (NLP) and machine learning. By harnessing the power of NLP algorithms, our framework aims to autonomously analyze and interpret resume content, enabling a deeper understanding of candidate qualifications beyond surface-level keywords. Coupled with machine learning techniques, our approach enables the system to adapt and refine its parsing accuracy over time, streamlining the screening process and enhancing the efficiency of candidate evaluation.

This research contribution aligns with ongoing efforts in the field of recruitment technology, offering a novel approach to resume parsing that transcends the limitations of traditional methods. By integrating NLP and machine learning, our framework promises to revolutionize the way recruiters extract, analyze, and evaluate resume data, ultimately expediting decision-making and ensuring a more comprehensive assessment of candidate suitability. The anticipated results of our study include a robust resume parsing framework capable of autonomously extracting key information from resumes with heightened accuracy and efficiency, thereby empowering recruiters to make more informed hiring decisions in a timely manner.

1.1 Motivation

In today's fast-paced job market, the sheer volume of resumes flooding in for every job opening presents a formidable challenge for recruiters. Manual parsing methods are time-consuming and prone to human error, while simplistic keyword matching algorithms often overlook crucial details. This inefficiency not only burdens recruiters but also risks missing out on top talent. Our project seeks to address this pressing issue by harnessing the power of Natural Language Processing (NLP) and machine learning. By developing an intelligent resume parsing framework, we aim to revolutionize the recruitment process. This framework will leverage NLP techniques to comprehend the contextual meaning and semantics of resume content. Machine learning algorithms will then be employed to extract key information accurately and efficiently. By automating this process, recruiters can significantly reduce the time spent on manual screening, allowing them to focus their attention on evaluating candidates more thoroughly. Ultimately, our goal is to enhance the accuracy and efficiency of resume parsing, enabling recruiters to make well-informed decisions and uncover the best talent more swiftly in today's competitive job market.

1.2 Scope of the Project

The scope of this project encompasses the development and implementation of an intelligent resume parsing framework using NLP and machine learning techniques. This framework will be designed to analyze and interpret resume content to extract key information such as personal details, education, skills, and work experience. The project will involve the integration of advanced NLP algorithms to enable semantic understanding of resume data, as well as the implementation of machine learning models to enhance parsing accuracy and efficiency.

Additionally, the project will involve the creation of a user-friendly interface for recruiters to interact with the parsing framework, allowing for seamless resume submission, parsing, and presentation of extracted information. The framework will be designed to handle resumes in various formats, including PDF, Word, and plain text, and will be scalable to accommodate large volumes of resumes.

The scope also includes the evaluation of the framework's performance through extensive testing and validation using diverse datasets. This evaluation will involve assessing parsing accuracy, efficiency, and scalability, as well as comparing the framework's performance against existing resume parsing methods.

Overall, the scope of this project is to develop a comprehensive and efficient solution for automating the resume parsing process, ultimately streamlining recruitment workflows and enabling recruiters to make more informed hiring decisions.

1.3 Problem statement

In response to the challenges posed by the manual screening of resumes, this study endeavors to develop an intelligent resume parsing framework. Leveraging advancements in Natural Language Processing (NLP) and machine learning, the framework aims to autonomously extract pertinent information from resumes, enhancing both accuracy and efficiency in candidate evaluation. By analyzing resume content semantically and adapting to diverse formats, the framework promises to revolutionize recruitment processes. Ultimately, this initiative seeks to empower recruiters with a scalable solution that streamlines resume parsing, expedites decision-making, and ensures a more comprehensive assessment of candidate qualification.

2. Literature Survey

Automated Resume Parsing: A Natural Language Processing Approach [1].The automated resume parsing system used a combination of data preprocessing, NER models, and Regex-based extraction to accurately identify and extract information from resumes. A diverse, annotated dataset was preprocessed to ensure uniformity and split into training and testing sets. SpaCy's NER model was trained to recognize entities like names, phone numbers, skills, and work experience. Regular expressions were developed for precise extraction of specific details, and NLP techniques enhanced context understanding. The combined results were filtered and displayed for easy review. This system proved accurate, efficient, flexible, and scalable, although it depended on high-quality training data and involved complex Regex development. Despite potential errors with unconventional formats, it effectively supported faster candidate evaluation and improved talent acquisition.

A Hybrid Resume Parser and Matcher using RegEx and NER [2].The proposed system employs a hybrid approach combining rule-based methods (regex) and Named Entity Recognition (NER) using NLP to extract information from resumes. Key attributes like Name, Phone number, Email, etc., are extracted using a combination of regex and Spacy's pre-trained transformer model. For resume scoring, the system calculates cosine similarity between the resume and job description using SBERT for vectorization. While the system is designed as an end-to-end web application with REST APIs for easy integration, the lack of dataset details makes assessing its generalizability challenging. Additionally, the absence of quantitative performance metrics or comparisons to existing approaches and the undisclosed accuracy of individual information extraction tasks are limitations. Reported parsing accuracy is 69.28%, with higher accuracy for email and phone number extraction but lower for experience and organization. Though the cosine similarity-based scoring approach is claimed effective, quantitative results supporting this claim are not provided.

Layout Aware Resume Parsing Using NLP and Rule-based Techniques [3]. The existing resume parsing system uses a combination of heuristic methods and machine learning techniques. It segments resumes into sections like profile, education, work experience, and skills using keyword matching, layout analysis, and predefined templates. Machine learning algorithms, particularly Conditional Random Fields (CRF), are trained on annotated datasets to extract information from each section. Named Entity Recognition (NER) further identifies specific entities such as names, contact information, and skills. The dataset for training and testing includes a wide variety of annotated resumes with different formats and layouts. While the system is accurate and efficient, capable of handling various resume formats, it relies heavily on heuristic criteria, which may cause errors with non-standard resumes. Performance can decline with highly customized layouts, and the system requires a large annotated dataset for training. Despite these challenges, the system shows satisfactory accuracy in extracting structured information, as indicated by evaluation metrics like precision, recall, and F1-score. Improvements are needed to enhance its robustness and adaptability.

Intelligent Recruitment System Using NLP [4]. The intelligent recruitment system uses NLP techniques to process and analyze resumes. It starts by extracting text from PDF resumes using PDFMiner, followed by data cleaning to remove stopwords and stem tokens. The cleaned data is

vectorized using CountVectorizer and TfidfTransformer from sklearn, transforming it into a numerical form. K-Means clustering categorizes the skills into Beginner, Intermediate, and Advanced levels, with the number of clusters determined by the Elbow Method. The system then predicts skill levels for new data and ranks candidates based on cosine similarity between their skills and the job description. The dataset includes various resume formats and job descriptions. This approach efficiently extracts and parses text, categorizes skills for effective evaluation, and ranks candidates by job fit. However, it depends on data quality, may oversimplify skill categorization, faces scalability challenges, and could introduce bias.

Improved Resume Parsing based on Contextual Meaning Extraction using BERT [5]. The improved resume parsing system employs contextual meaning extraction using BERT to enhance classification accuracy. Initially, resumes are collected from various sources like job portals, company websites, and social media platforms. These resumes undergo preprocessing steps, including tokenization, stopword removal, and normalization, to clean the text. Feature extraction methods such as TF-IDF, Bag of Words, or Word Embeddings are used to extract relevant information like skills, education, and experience. Machine learning algorithms like Naive Bayes, SVM, or Random Forest are then trained on these features to classify resumes according to job requirements. The performance of the classification model is evaluated using metrics such as accuracy, precision, recall, and F1-score. The dataset comprises resumes from job seekers, containing details about their skills, education, work experience, and contact information, often labeled for supervised learning. While the system efficiently handles large volumes of resumes and can be customized for specific job requirements, it relies heavily on keyword matching, which may lead to inaccurate classifications and lacks the ability to fully grasp contextual meanings and nuances in language. Manual feature engineering is also time-consuming. Despite achieving reasonable performance, there is room for improvement, particularly in understanding contextual meanings and enhancing classification accuracy, prompting the use of BERT vectorization for better results.

Resume Parsing Framework for E-recruitment [6].The paper introduces a resume parsing framework for e-recruitment, comprising three modules: document preprocessing, entity extraction, and resume analysis. Document preprocessing involves converting resumes into a machine-readable format, performing text extraction, language detection, and normalization. Entity extraction uses natural language processing to identify key information such as personal details, work experience, skills, education, and certifications. The resume analysis module computes metrics to evaluate candidate qualifications. The dataset consists of 1,000 resumes in various formats and languages.The framework is versatile, handling different resume formats and languages, and can be integrated into existing e-recruitment systems. However, the paper lacks detailed performance metrics and a thorough evaluation of the framework's generalizability. Additionally, it does not compare the proposed method with other existing resume parsing approaches. While the framework accurately extracts key information and provides insights to recruiters, these results are not quantified. A comparative analysis would enhance the understanding of its contributions.

Automated Resume Evaluation System using NLP [7]. The proposed system comprises three main modules: Conversion Phase, Extraction Phase, and Filtration and Ranking Phase. The Conversion Phase utilizes NLP techniques to convert unstructured resume data into structured data, while the Extraction Phase employs Named Entity Recognition (NER) to extract relevant information. The

Filtration and Ranking Phase filters and ranks resumes based on a comparison with job requirements. However, the paper lacks details on the dataset used for training and testing the system. While the automated resume evaluation system promises to reduce recruiters' time and effort, the absence of quantitative results and performance metrics for the system's modules is a notable limitation. Additionally, the generalizability of the NER model and a thorough comparison with existing resume screening methods are not evaluated. Despite presenting architectural diagrams and a system overview, specific outcomes of the implemented system are not provided.

Automated Resume Parsing and Job Domain Prediction using Machine Learning [8]. The automated resume parsing and job domain prediction system uses machine learning to streamline resume analysis. It preprocesses resumes with cleaning, tokenization, and lemmatization, and employs Named Entity Recognition (NER) using regular expressions and the spaCy library. A multi-class Support Vector Machine (SVM) algorithm predicts job domains from the extracted entities. Trained on a dataset of 1,000 manually annotated resumes from various fields, the system achieves a prediction accuracy of 92.08% and an F1-score of 0.92. This approach automates resume screening, reducing recruiters' workload and enhancing candidate selection. While it relies on manually labeled data and needs further research for improvement, it outperforms state-of-the-art methods, demonstrating its effectiveness in resume analysis and job matching.

End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT [9].The existing system efficiently parses and ranks resumes using advanced techniques. LinkedIn resumes are parsed with 100% accuracy by extracting text and metadata using pdftohtml and Apache Tika, then applying heuristic rules based on document structure. For non-LinkedIn resumes, it extracts text similarly and uses BERT for sequence classification, achieving 97% accuracy in converting them to LinkedIn format sections. Candidate ranking is based on the similarity between job descriptions and past job experiences, using BERT sentence pair classification with 72.77% accuracy. The dataset includes 715 LinkedIn and 1,000 non-LinkedIn resumes. The system excels in accurately parsing LinkedIn resumes and effectively utilizing BERT for classification and ranking. However, it relies on heuristic rules for non-LinkedIn resumes and lacks a ground truth dataset for ranking evaluation, posing some limitations. Despite these, it successfully differentiates and structures resumes, providing promising results in candidate suitability ranking.

Natural Language Processing and Text Mining to Identify Knowledge Profiles for Software Engineering Positions: Generating Knowledge Profiles from Resumes [10]. The system utilizes NLP and text mining to analyze resumes and generate knowledge profiles for software engineering positions. Using the Stanford CoreNLP library for tasks like tokenization and named entity recognition, it employs rule-based techniques to identify technical knowledge (TK) from a predefined list. Implemented as a web application, it follows a 4-stage text mining model. Tested on 40 resumes, the system achieved 98.1% accuracy with an F-measure of 90%, identifying common TKs like Java and JavaScript. While it automates resume analysis and speeds up the review process, it requires an expanded knowledge base and manual updates for new rules and TKs. The system is scalable and offers a structured way to assess candidate skills, despite needing improvements in accuracy and recall.

Resumate: A Prototype to Enhance Recruitment Process with NLP based Resume Parsing [11].The

proposed work details a methodology for resume parsing utilizing natural language processing and machine learning. Text preprocessing involves cleaning (lowercasing, removing Unicode characters, stop words, mentions, hashtags, links, and punctuation), tokenization, stop word removal, stemming, and lemmatization. Information extraction employs regular expressions and named entity recognition (NER) for names, phone numbers, emails, and skills. Multiple machine learning models, including KNN, SVM, Decision Trees, Naive Bayes, and XGBoost, were trained and evaluated using 5-fold cross-validation. XGBoost achieved superior performance in accuracy, precision, recall, and F1-score. Despite exploring various techniques and comparing model performances, the paper lacks specifics about the dataset and comprehensive comparisons with existing methods. It highlights the effective use of XGBoost for text classification but provides limited details on preprocessing and feature engineering steps.

Resume Parser with Natural Language Processing [12]. The resume parser uses NLP and text mining techniques to automate key information extraction and candidate-job matching. It converts PDF and DOC files to text using PyMuPDF and python-docx, respectively. Named Entity Recognition (NER) with tools like Stanford NER or Spacy identifies names and designations, while regular expressions extract details such as university names, degrees, skills, experience, phone numbers, and email addresses. The system uses scikit-learn for feature extraction and cosine similarity to compare resumes with job descriptions, determining their similarity. Tested on a dataset of 200 resumes from GitHub, the parser effectively extracts relevant details and calculates similarity percentages, aiding in candidate selection. Despite its efficiency, the system has limitations in extracting certain data points and potential biases in resume evaluation, indicating areas for further improvement.

Resume parsing using NLP [13]. The resume parser uses NLP techniques to convert and analyze resumes, ranking them according to company requirements. It employs OCR to handle different resume formats (PDF, Word, etc.) and includes a comprehensive NLP pipeline—lexical, syntactic, and semantic analysis, and Named Entity Recognition (NER) for domain-specific understanding (e.g., distinguishing "Python" the language from "python" the snake). Parsed data is stored in ElasticSearch, allowing for query-based scoring and ranking of resumes. The system automates resume parsing and ranking, efficiently handles various formats, and provides domain-specific term understanding. Results include a ranked list of applicants and a graphical dashboard for HR visualization. No specific limitations or cons were mentioned.

Resume Validation and Filtration using Natural Language Processing [14]. The resume validation and filtration system utilize NLP techniques for parsing and ranking resumes based on job requirements. It employs section-based segmentation for data extraction, extracting relevant information like name, email, phone number, experience, and skills. Comparing extracted data with job descriptions using cosine similarity, it ranks resumes accordingly. The system automates screening, reduces manual effort and bias, and allows candidates to verify and modify extracted data. Results include structured data presentation to candidates and a ranked list for recruiters, though specific limitations or cons were not mentioned.

Web Application for Screening Resume [15].The paper presents a web application for screening and rating resumes based on job requirements, featuring three main modules: Job Applicant Side (for resume upload, text extraction, and editing), Server Side (housing a training module for an NLP model

and a module for job description conversion), and Recruiter Side (calculating scores for resumes based on job requirements). The dataset includes resumes converted to JSON format using Dataturks, but specifics are not detailed. The application aims to reduce recruiters' workload by automating resume screening with NLP, allowing candidate edits for accuracy, and ranking candidates via a scoring formula. While the NLP model accuracy is reported as satisfactory, quantitative results and performance metrics are absent. The generalizability of the model and a detailed comparison with existing methods are not evaluated. Although the paper discusses limitations of other resume screening applications, a comprehensive comparative analysis is missing, which would have better contextualized the proposed system's contributions.

3. Design

3.1 High Level Design

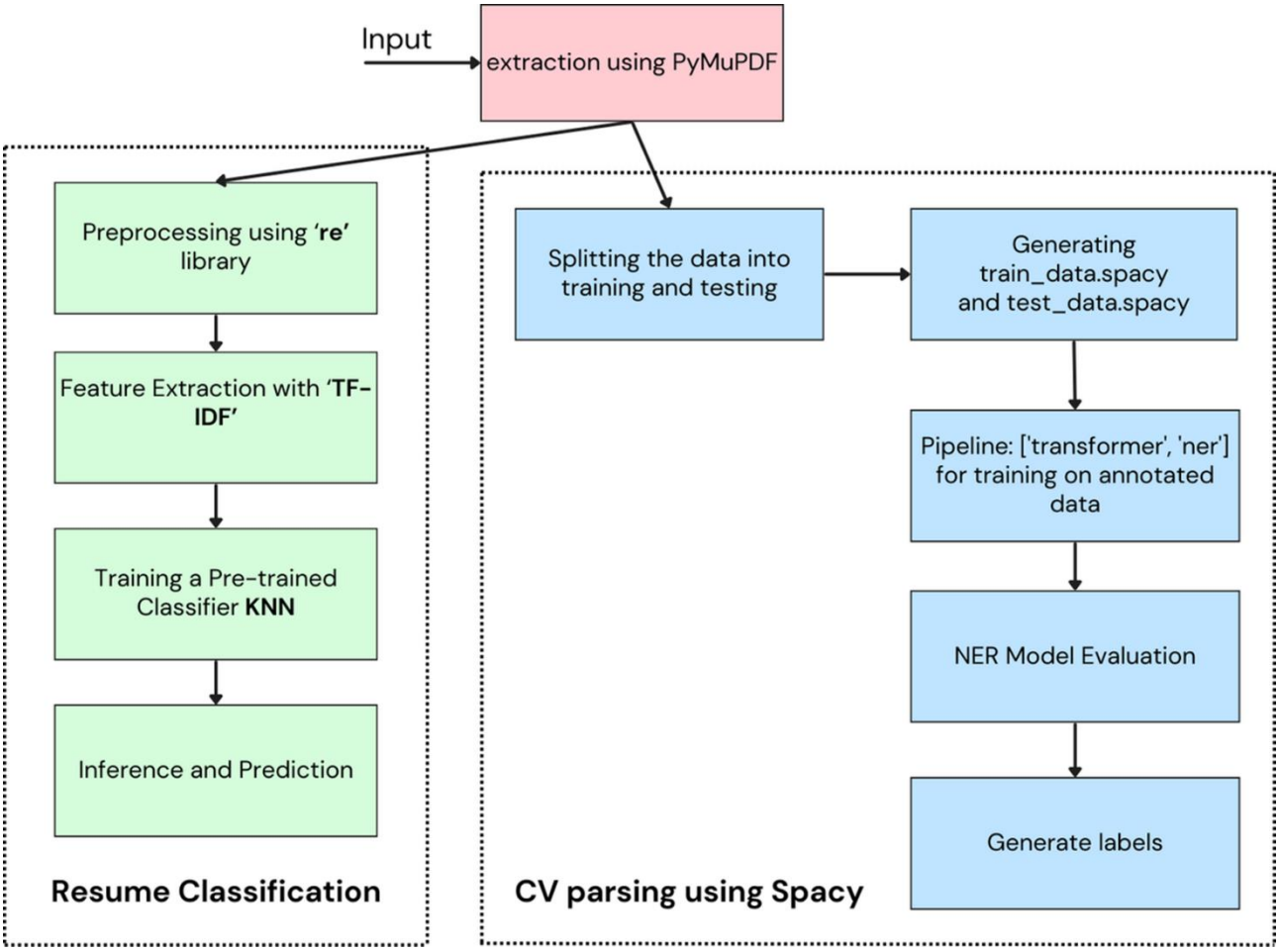


Fig 3.1: High level Design for CV parsing using Spacy

The process begins by extracting text from input PDF resumes using the PyMuPDF library. The extracted text undergoes preprocessing with regular expressions for cleaning and normalization. Feature extraction is then performed on the preprocessed text using the TF-IDF technique to evaluate word importance. A pre-trained K-Nearest Neighbors classifier is trained on this data for resume classification tasks. The data is also split into training and testing sets, converted to spaCy format, and used to train a pipeline consisting of a transformer model and named entity recognition (NER) model. This NER model is evaluated and then employed to generate labels and extract relevant information from the input resumes. The spaCy library is utilized for CV parsing, likely leveraging the trained NER model. The entire workflow combines PDF extraction, text preprocessing, machine learning classification, NER model training, and information extraction to automatically analyze and classify resume content.

3.2 Detailed Design

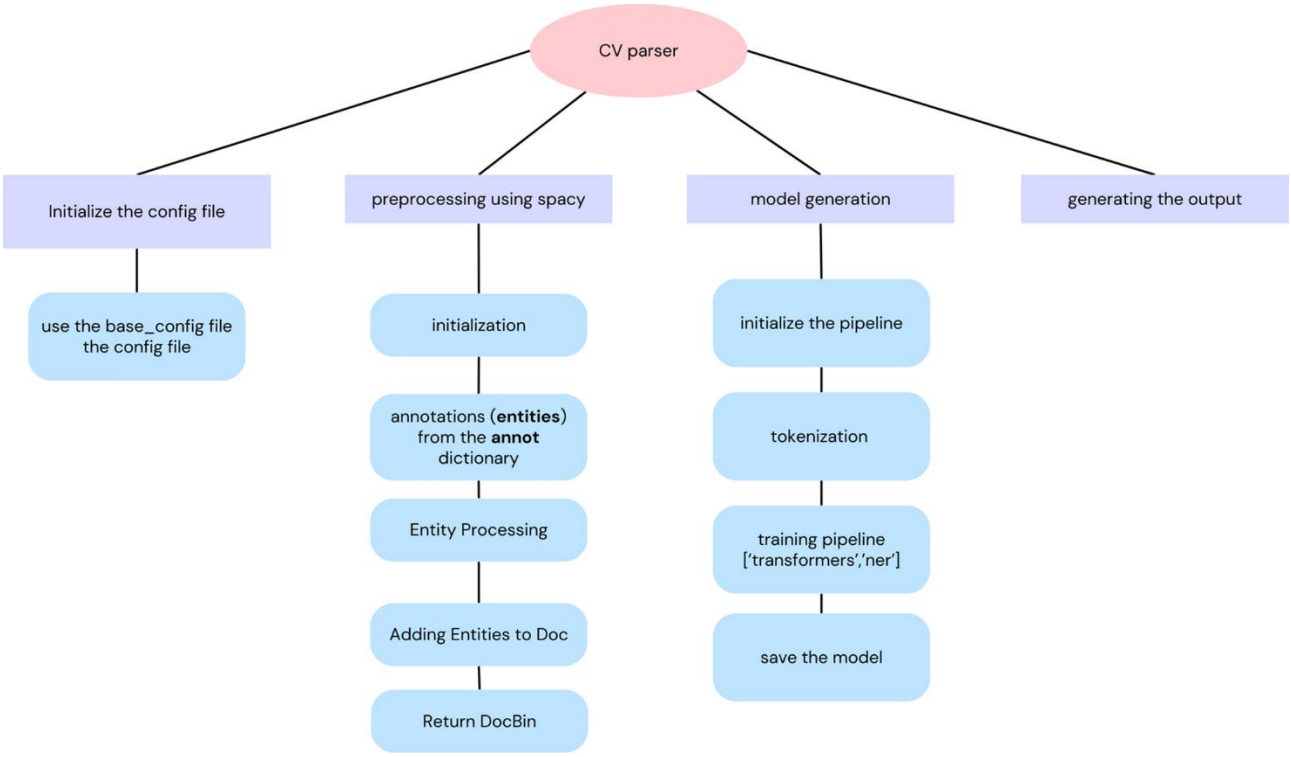


Fig 3.2: Detailed design for CV resume parser

The process begins by initializing a configuration file and using spaCy for preprocessing the input CV data. An initialization step loads annotations and entities from a predefined dictionary. Entity processing identifies relevant entities like names, dates, and skills in the text, which are then added to the spaCy document object. This processed data is returned in spaCy's binary format (DocBin). Next, a custom named entity recognition (NER) model is generated using spaCy's training pipeline, which involves tokenization and other preprocessing steps. The pipeline employs transformer-based neural networks to train the NER model on the annotated CV data. The trained model is then saved. Finally, this NER model is utilized to parse and extract structured information from new CV documents, generating the desired output.

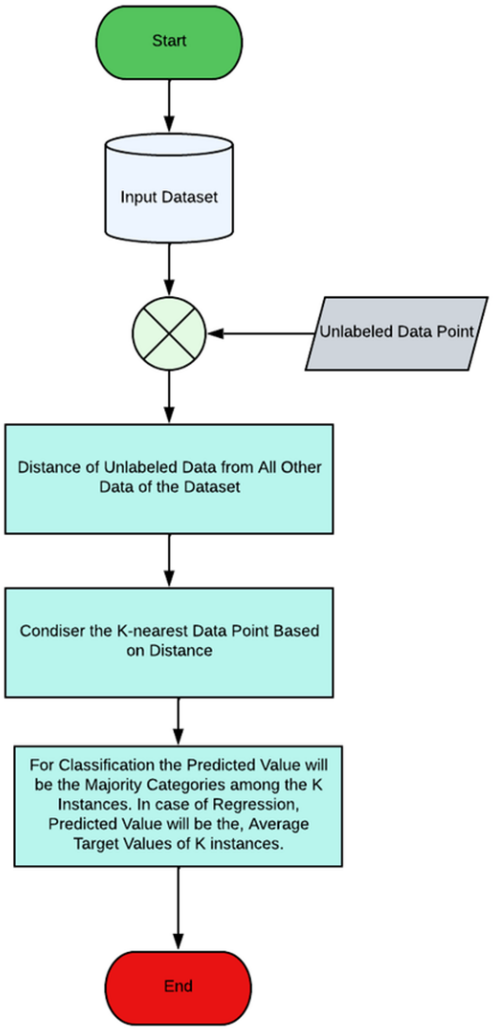


Fig 3.3: KNN algorithm workflow

The diagram depicts the workflow of the K-Nearest Neighbors (KNN) algorithm, which is widely used for classification and regression tasks. Beginning with a dataset containing both labeled and unlabeled data points, the algorithm calculates the distance between each unlabeled point and all other points in the dataset. It then identifies the K-nearest neighbors to the unlabeled point based on these distances. For classification tasks, the algorithm assigns the majority class label among these neighbors as the predicted label for the unlabeled point. In regression tasks, the predicted value is determined as the average of the target values of the K-nearest instances. The process concludes after assigning the predicted label or value to the unlabeled data point. In essence, the KNN algorithm leverages the proximity of labeled instances to unlabeled points to make predictions, making it a versatile and intuitive approach in machine learning.

4. Implementation

This section outlines the complete details of the solution for resume domain classification and label extraction, considering the modules implemented. The proposed solution leverages machine learning and natural language processing (NLP) techniques to analyze resumes, classify their domains, and extract relevant labels.

4.1 Proposed methodology

The solution is designed to classify resumes into specific professional domains and extract key information using the following methodology:

- Data Handling:

    1. Loading Data: The solution starts with loading resume data, typically stored in JSON format. The json library is used for this purpose, allowing the data to be read into a Python dictionary or list of dictionaries.

    2. Preprocessing: Once the data is loaded, preprocessing is essential to clean and standardize the text. This involves removing special characters, URLs, and other non-informative elements using regular expressions and other string manipulation techniques provided by libraries like re and pandas.

    3. Splitting: The cleaned data is then split into training and testing sets using the train_test_split function from sklearn.model_selection. This ensures that the model can be trained and evaluated on separate datasets.

- NER Training Pipeline

    1. Training: The custom Named Entity Recognition (NER) model is trained using the spaCy library. This involves creating a blank NLP pipeline, adding an NER component, and training it on annotated resume data.

    2. Customization: The NER model is customized to recognize entities specific to resumes, such as job titles, skills, and education.

- NER Model Evaluation

    1. Assessment: The performance of the trained NER model is assessed using spaCy's built-in scoring mechanism and custom evaluation scripts. This involves scoring the model against a test set and calculating metrics such as precision, recall, and F1-score.

2. Validation: Validation metrics provide insights into the accuracy and robustness of the NER model.

- Document Parsing

    1. Extraction: We use the PyMuPDF library to parse and extract text data from PDF resumes. This allows us to extract raw text content from resume files for further processing.

    2. Preprocessing: Text preprocessing techniques, such as regular expressions (re) and string manipulation, are applied to clean and standardize the extracted text data. This ensures consistent input for the classification and NER models.

- Resume Classification

    1. Categorization: We utilize TF-IDF vectorization from sklearn.feature_extraction.text to transform resume text into numerical features. This numerical representation is then used as input to the KNeighborsClassifier for classification.

    2. Representation: The numerical representation of resumes is fed into the KNeighborsClassifier for classification. The OneVsRestClassifier wrapper is used to handle multi-label classification scenarios.

    3. Prediction: The trained classifier predicts the job roles for new resumes based on their textual content.

4.2 Algorithm used for implementation

The algorithm used for the implementation involves two main components:

- **Named Entity Recognition (NER) using spaCy**: Named Entity Recognition is performed using a machine learning-based approach provided by the spaCy library. The algorithm involves training a custom NER model on annotated resume data. This model is trained to recognize entities such as skills, education, experience, etc., in resume text.

- **Domain Classification using K-Nearest Neighbors (KNN):** Domain classification is performed using the K-Nearest Neighbors algorithm implemented through scikit-learn. In this approach, resumes are represented as numerical feature vectors using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization. Then, a KNN classifier, wrapped in a OneVsRestClassifier, is trained on the TF-IDF vectors along with their corresponding domain labels. During classification, the KNN algorithm calculates the distance between a given resume vector and its k nearest neighbors in the feature space, and assigns the label based on the majority class among these neighbors.

4.3 Tools and technologies used

- **Flask:** A lightweight WSGI web application framework used to create the web interface for uploading and analyzing resumes.
- **scikit-learn:** A machine learning library in Python used for training and loading the classifier and TF-IDF vectorizer.
- **SpaCy:** An advanced natural language processing library in Python used for training the NER model and extracting labels from the resume text.
- **PyMuPDF:** A library for extracting text from PDF files.
- **Regular Expressions (re):** Used for text cleaning and preprocessing.
- **pandas:** Used for data manipulation and preprocessing.
- **json:** Used for loading resume data from JSON files.

4.4 Testing

Unit Testing:

- **cleanResume:** The function responsible for preprocessing resume text underwent rigorous unit testing with various input samples. Testing revealed that it effectively removes special characters, URLs, and non-ASCII characters while preserving essential information.
- **classification:** Unit tests conducted on the classification function demonstrated its accuracy in categorizing resumes into specific professional domains based on textual content. The function consistently produced the expected results across diverse resume samples.
- **extract_text_from_pdf:** Unit testing of the PDF text extraction function confirmed its reliability in extracting text data from resumes in PDF format. The function handled different PDF structures and layouts effectively, ensuring no loss of information during the extraction process.
- **SpaCy NER model:** The Named Entity Recognition (NER) model was thoroughly tested with various resume texts to validate its ability to identify named entities such as skills, education, and experience. The model exhibited high precision and recall in recognizing entities across different resume formats.

Integration Testing:

Integration testing was performed to validate the seamless integration of individual components into the web application's pipeline. The entire process, from resume upload to domain classification and label extraction, was tested comprehensively. Integration testing confirmed that all components interacted seamlessly, with correct output generation at each stage of the pipeline.

5. Results and Discussion

Domain Classification results:

Classifier	Accuracy
KNeighbors + OneVsRest	98.45%

NER Model Evaluation Results:

Metric	Score
Precision	74.47%
Recall	100.00%
F1-score	85.37%

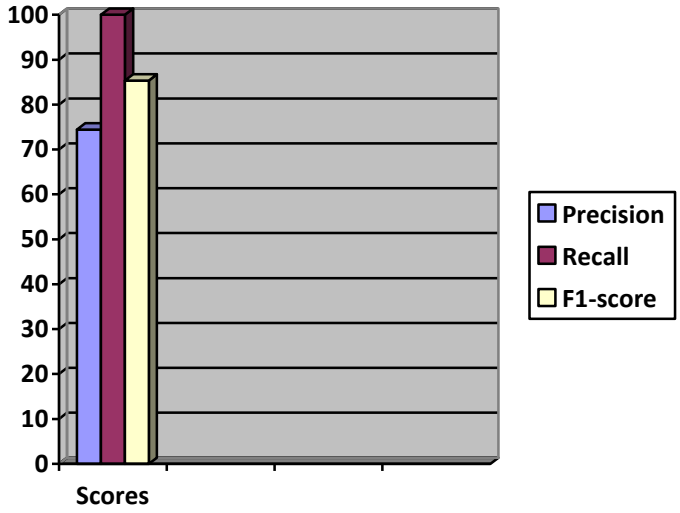


Fig 5.1: Graph depicting the precision and F1-scores

The KNeighborsClassifier achieved a high accuracy of 98.45% in classifying resumes into specific domains. However, the NER model, while achieving a perfect recall of 100%, exhibited lower precision and F1-score. This indicates that while the NER model effectively identifies relevant entities in resumes, there is room for improvement in terms of reducing false positive detections.

Resume Analysis Web Application UI:

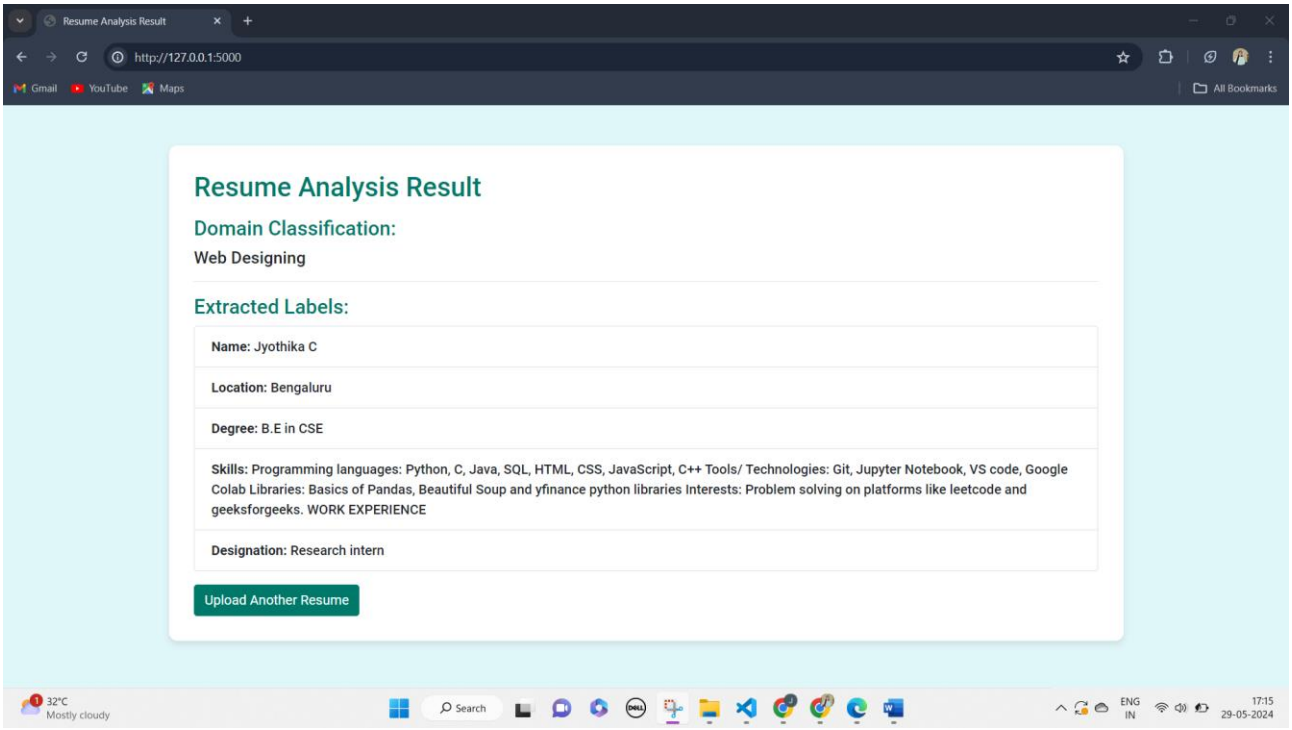


Fig 5.2: UI for Resume Analysis

In fig 5.2 a heading introduces the purpose of the page. Users are encouraged to explore domain detection.

Resume Upload Form:

- A heading ("Upload Your Resume") precedes the form.
- The form allows users to select a PDF file (resume) for analysis.
- The "Upload" button triggers the submission.

Error Handling:

- If there's an error (e.g., invalid file format), an alert message is displayed.

Resume Analysis Web Application UI:

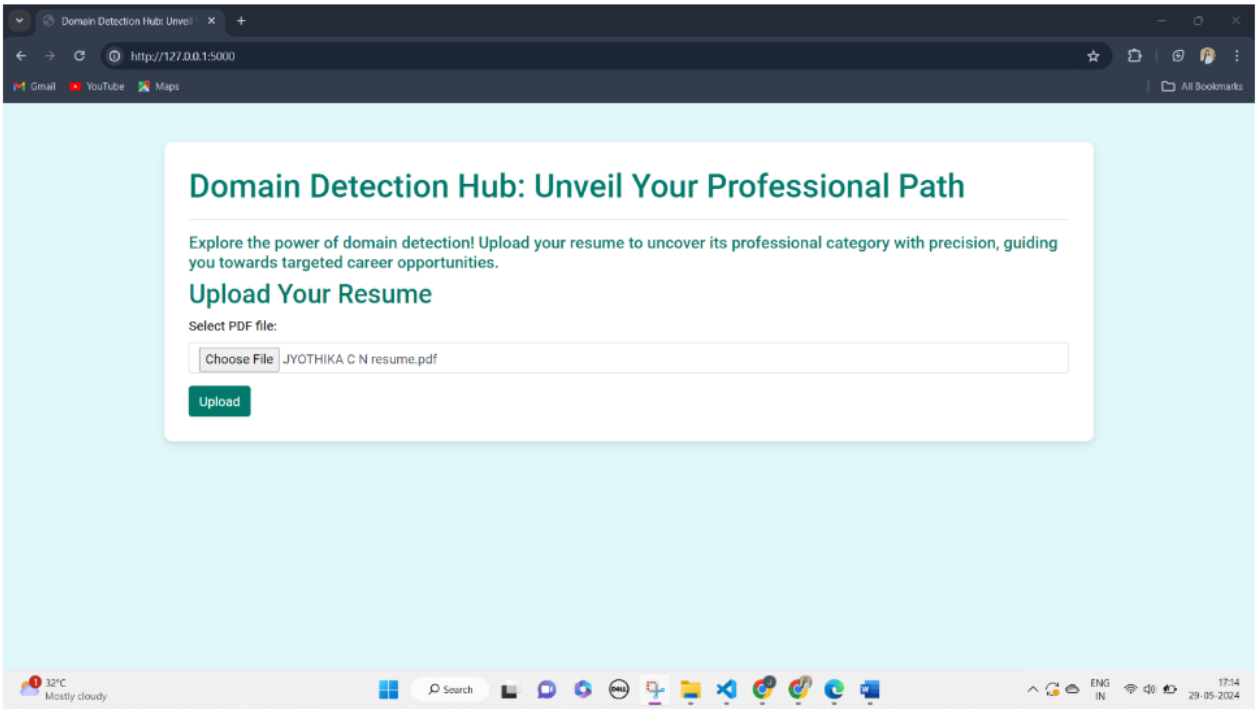


Fig 5.3:UI showing resume analysis result

The content is organized into two main sections:

- Domain Classification: This section displays the domain associated with the resume (e.g., technology, finance).
- Extracted Labels: Here, specific labels (such as skills, certifications) extracted from the resume are listed.

Finally, there's a button to upload another resume. This HTML page aims to present resume analysis results in a clean and user-friendly format.

6. Conclusion and Future Work

In our study, we presented a robust approach to resume analysis focusing on domain classification and Named Entity Recognition (NER). Our findings revealed a high accuracy of 98.45% in domain classification, underscoring the effectiveness of our model in categorizing resumes into specific professional domains. The NER model exhibited a commendable recall of 100%, albeit with room for improvement in precision and F1-score. Our work underscores the significance of accurate resume analysis in various applications such as job matching and talent acquisition. Moving forward, enhancing the precision of the NER model through data augmentation and ensemble learning techniques could mitigate the impact of false positive detections, further refining the accuracy and performance of the system. By addressing these shortcomings and incorporating proposed enhancements, we aim to advance research and applications in resume analysis, ultimately contributing to more efficient talent management processes.

References:

[1] T. G. Sougandh, S. S. K, N. S. Reddy and M. Belwal, "Automated Resume Parsing: A Natural Language Processing Approach," 2023 7th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 2023, pp. 1-6, doi: 10.1109/CSITSS60515.2023.10334236.

[2] G. M. G. R, S. Abhi and R. Agarwal, "A Hybrid Resume Parser and Matcher using RegEx and NER," 2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT), Faridabad, India, 2023, pp. 24-29, doi: 10.1109/ICAICCIT60255.2023.10466097.

[3] S. P. Warusawithana, N. N. Perera, R. L. Weerasinghe, T. M. Hindakaraldeniya and G. U. Ganegoda, "Layout Aware Resume Parsing Using NLP and Rule-based Techniques," 2023 8th International Conference on Information Technology Research (ICITR), Colombo, Sri Lanka, 2023, pp. 1-5, doi: 10.1109/ICITR61062.2023.10382773.

[4] A. Sharma, S. Singhal and D. Ajudia, "Intelligent Recruitment System Using NLP," 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV), Gandhinagar, India, 2021, pp. 1-5, doi: 10.1109/AIMV53313.2021.9670958.

[5] V. V. S. Tallapragada, V. S. Raj, U. Deepak, P. D. Sai and T. Mallikarjuna, "Improved Resume Parsing based on Contextual Meaning Extraction using BERT," 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2023, pp. 1702-1708, doi: 10.1109/ICICCS56967.2023.10142800.

[6]H. Sajid et al., "Resume Parsing Framework for E-recruitment," 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Korea, Republic of, 2022, pp. 1-8, doi: 10.1109/IMCOM53663.2022.9721762.

[7] R. Nimbekar, Y. Patil, R. Prabhu and S. Mulla, "Automated Resume Evaluation System using NLP," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India, 2019, pp. 1-4, doi: 10.1109/ICAC347590.2019.9036842.

[8] Sinha AK, Akhtar MAK, Kumar M (2023) Automated Resume Parsing and Job Domain Prediction using Machine Learning. Indian Journal of Science and Technology 16(26): 1967-1974. https://doi.org/10.17485/IJST/v16i26.880

[9] End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT. Vedant Bhatia, Prateek Rawat, Ajit Kumar, Rajiv Ratn Shah. Available at: https://arxiv.org/abs/1910.03089

[10] R. Valdez-Almada, O. M. Rodriguez-Elias, C. E. Rose-Gomez, M. D. J. Velazquez-Mendoza and S. Gonzalez-Lopez, "Natural Language Processing and Text Mining to Identify Knowledge Profiles for Software Engineering Positions: Generating Knowledge Profiles from Resumes," 2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT), Merida, Mexico, 2017, pp. 97-106, doi: 10.1109/CONISOFT.2017.00019.

[11] S. Mohanty, A. Behera, S. Mishra, A. Alkhayyat, D. Gupta and V. Sharma, "Resumate: A Prototype to Enhance Recruitment Process with NLP based Resume Parsing," 2023 4th International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2023, pp. 1-6, doi: 10.1109/ICIEM59379.2023.10166169.

[12]Resume Parser with Natural Language Processing, https://www.researchgate.net/publication/313851778_Resume_Parser_with_Natural_Language_Processing.

[13] https://zkginternational.com/archive/volume8/RESUME-PARSING-USING-NLP.pdf

[14]M. Alamelu, D. S. Kumar, R. Sanjana, J. S. Sree, A. S. Devi and D. Kavitha, "Resume Validation and Filtration using Natural Language Processing," 2021 10th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON), Jaipur, India, 2021, pp. 1-5, doi: 10.1109/IEMECON53809.2021.9689075.

[15]S. Amin, N. Jayakar, S. Sunny, P. Babu, M. Kiruthika and A. Gurjar, "Web Application for Screening

Resume," 2019 International Conference on Nascent Technologies in Engineering (ICNTE), Navi Mumbai, India, 2019, pp. 1-7, doi: 10.1109/ICNTE44896.2019.8945869.

APPENDIX:

DrillBit

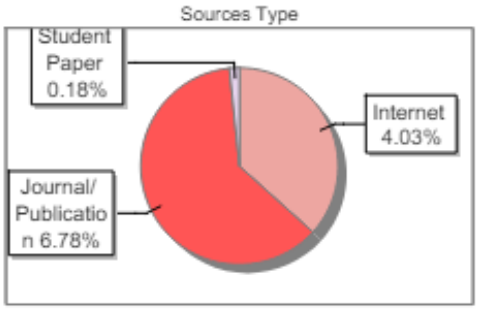
The Report is Generated by DrillBit Plagiarism Detection Software

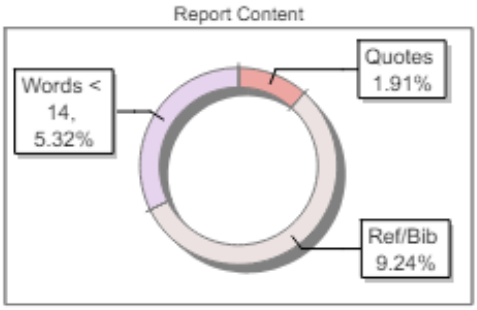
Submission Information

Author Name	Jyothika
Title	C N
Paper/Submission ID	1902148
Submitted by	hod.cse@bmsce.ac.in
Submission Date	2024-05-30 11:36:52
Total Pages, Total Words	24, 5455
Document type	Project Work

Result Information

Similarity **11 %**





Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	0 %
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File