Grenze International Journal of Engineering and Technology, June Issue

Resume Shortlisting based on the Job description for Academic Institution using NLP

Rakhi Wajgi[1], Arpita Pimparkar[2], Astha Bhiwapurkar[3], Khushalee Wande[4], Maitreyee Warhadpande[5] and Samiksha Raut[6]
[1-6]Yeshwantrao Chavan College of Engineering/ Computer Technology, Nagpur, India
Email: rakhiwajgi17@gmail.com, arpitaapimparkar@gmail.com, asthabhiwapurkar@gmail.com, khushaleewande11.@gmail.com, maitreyeewarhadpande27@gmail.com, samikshasr1704@gmail.com

***Abstract*—Manually shortlisting resumes is labor-intensive and inefficient, especially with large volumes, leading to inconsistencies. This research leverages NLP and Named Entity Recognition (NER) to extract key details like education, skills, and work experience from resumes, automating the shortlisting process. A system was developed to rank candidates based on job relevance, generating downloadable Excel files and providing insights through visuals. The proposed method significantly reduces manual effort and time, improving accuracy and efficiency in candidate selection.**

I. INTRODUCTION

Recruitment processes often involve analyzing job requirements and screening candidates, a task that becomes particularly challenging when handling a large volume of resumes. Academic institutions face a similar difficulty during campus recruitment, where recruiters must shortlist students who best match the job descriptions provided by visiting companies. Manually reviewing resumes is time-consuming and inefficient, especially when dealing with unstructured data in various formats.

To address these challenges, this paper proposes an intelligent resume shortlisting system tailored for educational institutions. The system enables users to upload multiple resumes in bulk and filter them based on predefined criteria through an intuitive interface. Utilizing Python libraries like NumPy, Pandas, Scikit-learn, and NLTK, the system processes resumes, extracts relevant information, and ranks candidates based on academic qualifications, experience, and skills. The results are presented in a downloadable Excel file along with a dashboard, streamlining the traditionally labor-intensive process of resume screening.

The proposed solution leverages Natural Language Processing (NLP) to convert unstructured resume data into a structured format, enabling accurate analysis. While current methods achieve high accuracy, challenges such as handling diverse resume formats and inconsistent terminology remain. A dataset of resumes from college students with varied academic backgrounds serves as the basis for training and evaluating the system, ensuring it meets the needs of institutions and recruiters effectively.

II. LITERATURE SURVEY

ML and NLP approaches make it possible to shortlist resumes in an efficient and objective manner. KNN, SVM and Random Forrest algorithms are used to enhance categorization accuracy in which one study found KNN to achieve 92% ranking accuracy.[5]

Grenze ID: 01.GIJET.11.2.300_20

According to this research paper[1] same scoring method based on cosine similarity was trained using Multinomial Naive Bayes classifier which claimed to produces useful ranking results. As per the study it claims that the Multinomial Naive Bayes classifier achieves an accuracy of about 73.97% for the classification task.
The study developed an automated system that can classify resumes utilizing SVM algorithms reaching one hundred percent accuracy alongside other methods aimed at classifier tests. Nevertheless it is dependent on training data which is an impediment to adapting to the job market requirements.[7]
The system's classifiers included the Linear SVM which achieved the highest accuracy of 78.53% for resume categorization. However, this only works with CSV files which is not commonly used and does not support regular files that are in a .doc or simply a .pdf format.[9]
Another approach to resume parsing is by utilizing NLP and deep learning which along with Resume Parser and Ranker claimed to achieve 93% NER accuracy while in a variety of different metrics showed more than 80% accuracy with incorporation of scoring bias and recognition of a diverse multitude of resume formats.[8]
This paper [3] introduces a dual-model approach combining a custom NER model and MLP for multi-label classification, improving text extraction and addressing ATS inefficiencies. The model achieved 78.6% accuracy, outperforming SRNN and LSTM in multi-class classification tasks.
The study explores the application of NLP in automating the resume shortlisting process, offering improved efficiency and reduced bias compared to traditional methods. It discusses the use of techniques such as feature extraction and machine learning, with recent innovations in transformer models enhancing accuracy and contextual comprehension. [2]
The paper explores the implementation of NLP to enhance the efficiency of applicant screening by automating the extraction and analysis of resume data. It discusses the use of machine learning models to improve accuracy and addresses challenges like handling varied resume formats and reducing biases in the screening process. [4]
This paper investigates the use of machine learning methods to automate resume classification, aiming to enhance the recruitment process's efficiency and accuracy. It discusses different ML models and techniques for extracting and categorizing resume information, emphasizing improvements in model effectiveness and their application in hiring. [6]
The paper explores the use of NLP techniques to automate the resume evaluation process, focusing on text parsing, feature extraction, and classification methods to improve the efficiency and precision of candidate selection.[10]
The paper presents a system called JRC that utilizes machine learning for classifying job postings and resumes in online recruitment. It focuses on the application of text classification and feature extraction techniques to improve the accuracy and efficiency of matching candidates with job roles, enhancing the overall recruitment process.[11]

III. BLOCK DIAGRAM

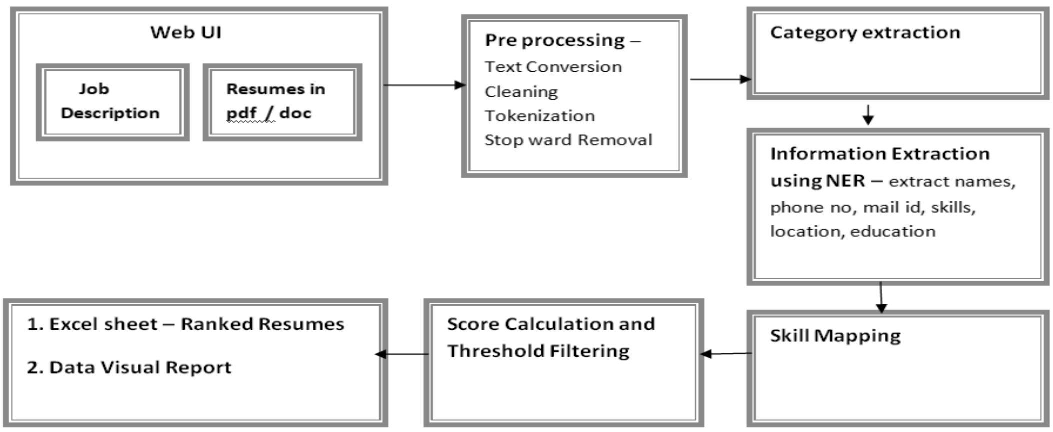

Fig 1: Resume Shortlisting Web Application Block Diagram

"Fig.1" Users can upload resumes, select criteria, and download shortlisted resumes. The resumes are parsed, with text extracted, cleaned, and tokenized before being stored in a Data Frame. Criteria are selected and stored, and a filtering module matches resumes to the criteria using a machine learning algorithm. The ranked resumes are then visualized with charts and exported to an Excel file.

IV. METHODOLOGY

The Resume Shortlisting system involves a structured process for processing resumes, extracting details, and matching them to job descriptions based on relevance, automating the shortlisting phase to streamline recruitment.

A. Dataset Used

The dataset used in the Resume Shortlisting System has been created by gathering resumes from final-year students, providing a rich source of real-world data for analyzing and matching candidates to job descriptions.

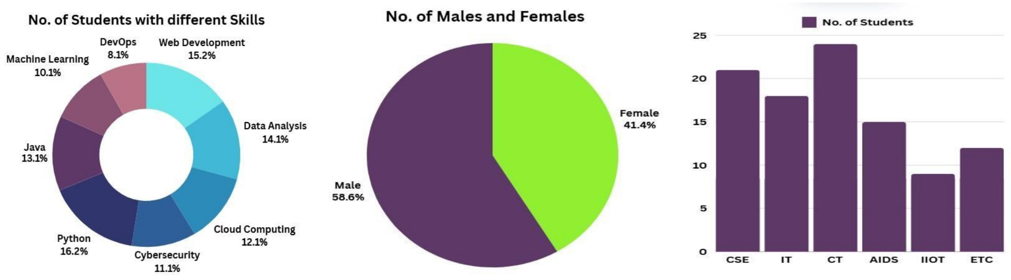

Fig 2. Overall statistics of Dataset

The dataset includes key information obtained from the student's resumes "Fig.2" as below:

- Information pertaining to identity: Full name, phone contact, and physical address.
- Academic Qualifications: Type of certificate obtained, school attended, year of completion, and GPA.
- Skills: Both hard and soft skills, and professional certificates earned.
- Projects: Summary of academic and/or personal projects including their objectives and results.
- Language: Different languages with different levels of fluency.

This dataset can be utilized to understand entry-level hiring practices as it contains a wide range of technical and soft skills mix, certifications, academic projects and details about academic history which is helpful in assessing candidates looking for internships and entry-level positions

B. Input and Initialization:

This phase ensures the foundation for the resume matching system is set with data, tools, and configurations:

- Threshold: Specifies the minimum match score (in percentage) a resume must achieve to qualify for shortlisting. This helps filter out less relevant resumes.
- Job Descriptions: Lists of strings outlining role requirements, e.g., "Looking for a Data Scientist proficient in Python, SQL, Machine Learning, and data visualization tools."

C. Pre-processing

Prepare resumes for analysis by cleaning and structuring their content.

- Text Extraction: Tools like PyPDF2, textract, python-docx, and OCR software are employed.
- Text Cleaning: Remove filler words, convert text to lowercase, remove punctuation, and standardize format using libraries like spaCy or NLTK.

D. Category Extraction and Matching

Assign resumes to relevant job categories and calculate how closely they match the provided job descriptions.

Category Assignment

Train a classification model, such as OneVsRestClassifier, using TF-IDF to convert text into numerical data.
Formula:

$$Category\ i = argmax(P(C_{text}))$$

Where P is the predicted probability of a category for the cleaned text.

Percentile Match Calculation:

Uses cosine similarity to compute the match between resume content and job descriptions:

$$Similarity(A, B) = \frac{A.B}{||A||||B||}$$

Where A and B are the vectorized forms of the resume and job description.
Match percentile:

$$Match\ Percentile = \left(\frac{Similarity\ Score}{Max\ Possible\ Score}\right) \times 100$$

TF-IDF captures keyword importance efficiently, while OneVsRestClassifier ensures scalability for multi-class classification. Relevance scoring compares resume text with job descriptions using TF-IDF similarity and cosine similarity with Word Embeddings. Alternative methods include Semantic Similarity using BERT or Sentence-BERT.

E. Information Extraction

Identify and extract key details from resumes for recruiter reference.

- Extract essential fields using a combination of regular expressions and Named Entity Recognition (NER).
- Use libraries like spaCy for NER tasks or predefined templates for specific data fields.

$$Extracted\ Skills = Skills\ Dictionary\ \cap C\_text$$

NER (fitz/PyMuPDF): In the Resume Shortlisting project, text is extracted from PDFs using PyMuPDF, and NER with spaCy identifies key details like Name, Email, Phone Number, Skills, and Education.The extracted entities can be represented as:

$$E = \{(e1, C1), (e2, C2), \ldots, (en, Cn)\}$$

The system uses spaCy's pre-trained NER models to extract and categorize entities (e.g., Name, Email, Skills) from resumes, facilitating accurate parsing and efficient matching with job descriptions for candidate shortlisting.

F. Threshold Filtering and Sorting

In the Resume Shortlisting project, resumes are ranked by match score using a threshold mechanism. Resumes with a score below the threshold are filtered out, and the remaining resumes are sorted in descending order of match score using Python's sorted() function. The threshold value τ defines the minimum match score required for relevance.
The threshold filtering process can be represented mathematically as:

$$R^{\wedge} = \{ri | score(ri) \geq \tau\}$$

- ri = each resume
- score(ri)= match score of rir_iri
- τ = threshold

G. Output Generation

Provide recruiters with a clear and organized summary of shortlisted resumes.
Compile the relevant data for each filtered resume, including:

- File name of the resume.
- Match score.
- Extracted information (name, education, skills, email, phone).
- Save the compiled data into an Excel file using Python libraries like pandas or openpyxl.
- Ensure the file is formatted for easy readability and usability by recruiters.

H. Data Analysis

A Visualization will be developed to provide recruiters with a clear, data-driven overview of the recruitment process. It visualizes important metrics, providing actionable insights into candidate performance and job matching.

Tools and Technologies used

- Text Processing: PyPDF2, textract, python-docx, Tesseract.
- Natural Language Processing: NLTK, spaCy, scikit-learn.
- Data Handling: pandas, openpyxl for working with Excel files.
- Similarity Scoring: TF-IDF, Percentile Ranking.

V. ALGORITHM

Algorithm 1: Process of Resume Extraction
Input: Resume directory containing PDF resumes A predefined taxonomy of skills **Output:** Processed data with extracted text, tokenized words, skills, and categories CSV/Excel files and a category distribution pie chart **for** each resume file in the file directory: Extract text using ***upload_resumes()*** **if** extraction is successful: 1. Clean the text using ***clean_text();*** 2. Tokenize and filter the text using ***tokenize_and_filter();*** *3.* Extract skills using ***extract_skills();*** *4.* Standardize skills using ***extract_skills();*** **if** extraction fails: Log the error for the failed file; Store results in a *DataFrame*; Save the DataFrame to both **CSV** and **Excel** formats; Analyze the processed dataset: Count the occurrences of each category; Create a pie chart: 1. Set the figure size and labels; 2. Display percentages; Display the generated pie chart;

VI. RESULTS

The system achieved a 95% accuracy for NER in matching resumes to job descriptions, demonstrating the effectiveness of the implemented matching approach based on percentile calculations and job-specific thresholds. (Table I)
The threshold successfully filtered resumes, ensuring only highly relevant ones were selected. This can assist recruiters in prioritizing top candidates. The percentile-based approach and threshold filtering are effective in automating and optimizing resume evaluation. The visualizations demonstrate the system's robust ability to analyze and present resume-match data effectively. The combination of bar charts, pie charts, word clouds, and histograms provides comprehensive insights into resume performance, skill trends, and overall match distribution.

Reason for misses

- Formatting issues: Skill mentioned in an unusual or non-standard format.
- Synonyms: Alternative phrasing (e.g./ "ML" instead of "Machine Learning")
- Incomplete Parsing: Text extraction failure due to complex resume layout.

TABLE I. RESUME RETRIEVAL

Skill	Total Resumes	Correctly Retrieved	Percent
Python	20	20	100%
Machine Learning	15	10	66.6%
NLP	12	12	100%
Data Analysis	9	7	77.77%
TensorFlow	5	4	80%

VII. CONCLUSION

Resume shortlisting is one of the most important aspects of recruitment as it simplifies the process of searching and filtering through a large number of applications where the right match is in high demand. The market has become more competitive than ever, and adoption of tools that save time in hiring is starting to become a priority for companies, utilizing a logical viewpoint that centres around core competencies, achievements and the right fit for the company. Automated tools and sharp selection criteria also contribute to the precision of shortlisting, so only the right people make it into the next phase. In conclusion, a well-controlled and supervised shortlisting process of resumes opens the doors to effective recruitment and employee retention. In future we will work on creating a complete solution for retrieving resumes that will start right from the upload with the use of Gen AI techniques and also will work on increasing retrieval accuracy.

REFERENCES

[1] P. M. Jacob, S. Jacob, J. Cheriyan and L. S. Nair, "ResumAI: Revolutionizing Automated Resume Analysis and Recommendation with Multi-Model Intelligence," 2023 Global Conference on Information Technologies and Communications (GCITC), Bangalore, India, 2023, pp. 1-7.

[2] S. Ambareesh, N. K. Thakur, U. Bhattarai, S. K. Yadav, J. N. Thakur, and A. K. Mahato, "Resume Shortlisting Using NLP," 2024 4th International Conference on Data Engineering and Communication Systems (ICDECS), Bangalore, India, 2024, pp. 1-5

[3] A. S. Charan and D. S. Moirangthem, "Leveraging Name Entity Recognition and Text Classification for Talent Mapping," 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2024, pp. 1-6

[4] B. Lalitha, S. Kadiyam, R. V. Kalidindi, S. M. Vemparala, K. Yarlagadda, and S. V. Chekuri, "Applicant Screening System Using NLP," 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Uttarakhand, India, 2023, pp. 379-383

[5] Tejaswini Ka, Umadevi Vb, Shashank M Kadiwal a, Sanjay Revannaa "Design and development of machine learning based resume ranking system", Global Transitions Proceedings 3 (2022) 371–375, Bengaluru 560 019, Karnataka, India

[6] B. Surendiran, T. Paturu, H. V. Chirumamilla, and M. N. R. Reddy, "Resume Classification Using ML Techniques," 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, 2023, pp. 1-5

[7] G. Kamineni, K. A. Sai and G. S. N. Rao, "Resume Classification using Support Vector Machine," 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 2023, pp. 91-96.

[8] B. Nisha, V. Manobharathi, B. Jeyarajanandhini and G. Sivakamasundari, "HR Tech Analyst: Automated Resume Parsing and Ranking System through Natural Language Processing," 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp. 1681-1686

[9] Kumar Roy, Sarabjeet Singh Chowdhary, Rocky Bhatia, "A Machine Learning Approach for automation of Resume Recommendation system", International Conference on Computational Intelligence and Data Science (ICCIDS 2019)

[10] 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,