A PROJECT REPORT
on

"AUTOMATED RESUME SCREENING SYSTEM"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE & ENGINEERING

BY

SHREYA ALLUPATI	2205673
ISHA DURGE	2205210
SAI MANIKANTA PATRO	22051009
ARIKATHOTA HRUDAY VIKAS	22052802
KUNJAL GROVER	2205297
MANCHALA GURU SAI ESHWAR REDDY	2205991

UNDER THE GUIDANCE OF
PROF. PRADEEP KANDULA

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

April 2025

A PROJECT REPORT
on

“AUTOMATED RESUME SCREENING SYSTEM”

Submitted to
KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR’S DEGREE IN
COMPUTER SCIENCE & ENGINEERING

BY

SHREYA ALLUPATI	**2205673**
ISHA DURGE	**2205210**
SAI MANIKANTA PATRO	**22051009**
ARIKATHOTA HRUDAY VIKAS	**22052802**
KUNJAL GROVER	**2205297**
MANCHALA GURU SAI ESHWAR REDDY	**2205991**

UNDER THE GUIDANCE OF
PROF. PRADEEP KANDULA

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA -751024 April 2025

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

“AUTOMATED RESUME SCREENING SYSTEM“

submitted by

SHREYA ALLUPATI	**2205673**
ISHA DURGE	**2205210**
SAI MANIKANTA PATRO	**22051009**
ARIKATHOTA HRUDAY VIKAS	**22052802**
KUNJAL GROVER	**2205297**
MANCHALA GURU SAI ESHWAR REDDY	**2205991**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci-ence & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during the year 2024-2025, under our guidance.

Date: / /

(Prof. Pradeep Kandula)
Project Guide

Acknowledgements

This group work is the culmination of the endeavors of its team members, who all strived to put in their best efforts to see this project to its fruition. We are grateful to the loyalty and dedication our team has shown in bringing out their best for the success of this project.

We are immensely thankful to our institute, Kalinga Institute of Industrial Technology, Bhubaneswar, for providing us this platform to display the fruits of our labour on this project.

We are profoundly grateful to **Prof. PRADEEP KANDULA** of **Affiliation** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. ....

SHREYA ALLUPATI
ISHA DURGE
SAI MANIKANTA PATRO
ARIKATHOTA HRUDAY VIKAS
KUNJAL GROVER
MANCHALA GURU SAI ESHWAR REDDY

ABSTRACT

The Automated Resume Screening System is designed to optimize the recruitment process by leveraging Natural Language Processing (NLP) techniques. This system efficiently evaluates resumes against job descriptions, minimizing manual effort and mitigating bias. It incorporates advanced models like DistilBERT and BERT for extracting skills and assessing experience, ensuring precise alignment between candidates and job requirements. Additionally, it employs cosine similarity with weighted scoring to rank resumes based on relevance, promoting a more objective selection process.

The system encompasses key functionalities such as resume parsing, skill matching, experience evaluation, and ranking. It supports multiple resume formats and generates insightful summaries using models like FLAN T5. Built with a scalable architecture, the solution is designed for cloud deployment. Future developments include expanding multilingual capabilities and integrating with Applicant Tracking Systems (ATS). Overall, this automated screening system streamlines the initial phases of hiring, enhancing both efficiency and accuracy in candidate selection.

Keywords: Automated Resume Screening, NLP Techniques, Resume Parsing, Skill Matching, Resume Ranking

Contents

List of Figures

Chapter 1

Introduction

In today’s highly competitive job market, recruiters encounter considerable difficulties in effectively screening and selecting candidates from a vast pool of applicants. Traditional resume screening methods are largely manual, making the process not only time-consuming but also susceptible to human bias. The growing number of job applications further complicates hiring, creating inefficiencies in identifying the best-suited candidates.

Many existing recruitment systems lack intelligent automation, often relying on basic keyword-based searches that may fail to recognize qualified candidates who use different phrasing in their resumes. Additionally, these systems struggle with accurately assessing experience, educational background, and skill relevance, resulting in mismatches between job requirements and shortlisted candidates.

To overcome these challenges, the Automated Resume Screening System utilizes Natural Language Processing (NLP) and Machine Learning (ML) to enhance both the accuracy and efficiency of candidate selection. By systematically analyzing resumes and ranking them based on job descriptions, this system reduces the burden on recruiters while ensuring fair, data-driven hiring decisions.

By adopting this AI-powered solution, organizations can streamline their recruitment process, significantly reducing time and effort while improving the overall quality of hires.

.

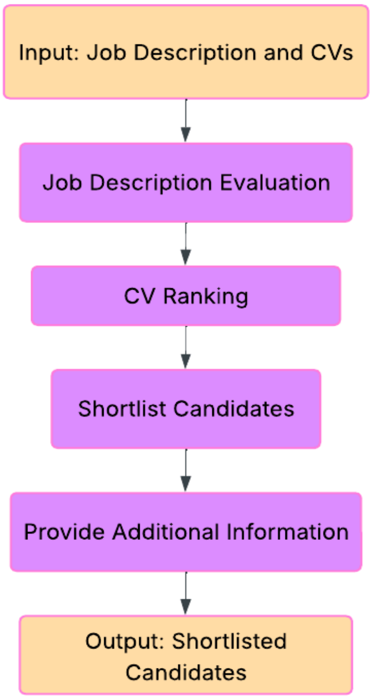

Input: Job Description and CVs
Job Description Evaluation
CV Ranking
Shortlist Candidates
Provide Additional Information
Output: Shortlisted Candidates

Figure 1.1: Automated Resume Screening Pipeline

Chapter 2

Basic Concepts/ Literature Review

This chapter introduces the core principles, methodologies, and technologies employed in building the Automated Resume Screening System. Additionally, it includes a review of existing literature to demonstrate how this project enhances recruitment efficiency and aligns with prior research in the field.

2.1 Resume Screening and Its Challenges

Resume screening is the first stage in the hiring process, where recruiters assess candidates' qualifications based on their submitted resumes. Conventional manual screening approaches face several limitations:

- Time-intensive – Reviewing numerous applications requires significant effort.
- Subject to bias – Human judgment can introduce unfair preferences.
- Lack of uniformity – Different evaluators may assess the same resume in varying ways.

To address these issues, Automated Resume Screening Systems utilize artificial intelligence technologies, including Natural Language Processing (NLP), Machine Learning (ML), and Information Retrieval (IR), to optimize and standardize the screening process.

2.2 Natural Language Processing (NLP)

2.2.1 Fundamentals of Natural Language Processing

Natural Language Processing (NLP), a branch of artificial intelligence, focuses on enabling machines to comprehend, process, and produce human language. In automated resume screening systems, NLP proves indispensable for parsing and evaluating textual content from both candidate resumes and job postings.

2.2.2 Key NLP Methods for Resume Analysis

Modern resume screening systems employ several NLP techniques:

- Tokenization: Segmenting text into individual words or meaningful phrases
- Named Entity Recognition (NER): Detecting and classifying critical information such as personal names, technical skills, professional roles, and company names
- Part-of-Speech Tagging: Analyzing grammatical components to understand resume content structure
- Text Similarity Analysis: Implementing algorithms like Cosine Similarity and Jaccard Similarity to match resumes with job requirements
- Keyword Identification: Utilizing advanced methods including TF-IDF, RAKE, and Transformer-based models to extract pertinent terms and phrases

2.3 Machine Learning for Resume Screening

2.3.1 Keyword and Skill Matching – DistilBERT/XLM

DistilBERT represents an optimized variant of the BERT architecture, offering an optimal balance between computational efficiency and model performance. This distilled model proves particularly effective for rapid identification of relevant keywords and competencies in resumes, with its contextual embeddings providing superior interpretation of domain-specific terminology.

The XLM framework extends these advantages to cross-lingual applications, enabling robust analysis of resumes across multiple languages. When benchmarked against conventional approaches like SpaCy's Named Entity Recognition and TF-IDF vectorization, both DistilBERT and XLM demonstrate:

- Enhanced semantic comprehension of resume content
- Improved accuracy in information extraction
- Better handling of nuanced professional terminology

2.3.2 Experience Quantification – BERT + Regex

BERT leverages transformer-based attention mechanisms to understand nuanced context in professional experience sections. Combined with Regex's pattern-matching strengths for structured data (dates, job titles, companies), this hybrid approach ensures robust extraction from both unstructured and formatted resume content.

Key Advantages:

- BERT provides deep contextual understanding of roles and responsibilities
- Regex precisely captures standardized information (employment periods, positions)
- Outperforms traditional systems (like SpaCy NER) in:
  - Handling diverse resume fomats
  - Maintaining accuracy with unconventional phrasing
  - Adapting to mixed-format documents

The integration delivers comprehensive data extraction while balancing contextual analysis with structured pattern recognition.

2.3.3 Resume Ranking – Cosine Similarity + Weighted Scoring

The system employs Cosine Similarity to quantitatively assess the alignment between:

- Distributed representations of resume content (generated through DistilBERT embeddings)
- Vectorized job description requirements

To further refine candidate evaluation, a Weighted Scoring mechanism enables recruiters to:

1. Prioritize specific qualification categories
2. Assign customized importance weights (e.g., Technical Skills: 40%, Professional Experience: 30%, Education: 20%, Certifications: 10%)
3. Adjust scoring parameters based on evolving hiring needs

This approach outperforms simple keyword-based methods (like TF-IDF) by considering the semantic meaning of resume content, resulting in more accurate rankings.

2.3.4 Summarization – FLAN-T5 (CPU)

FLAN-T5 Model:

- Provides efficient CPU-based operation
- Maintains satisfactory summarization quality
- Enables resource-constrained deployment
- Suitable for standard processing requirements

2.4 Information Retrieval for Resume Matching

2.4.1 Cosine Similarity for Resume Matching

The system employs cosine similarity as its core metric for quantifying the semantic alignment between candidate resumes and job descriptions. Mathematically, this measure calculates the cosine of the angle θ between two document vectors in a high-dimensional embedding space:

$$\cos(\theta) = (A \cdot B) / (||A|| \times ||B||)$$

Where:

- A, B = Vector representations of documents (resume and JD respectively)
- $A \cdot B$ = Dot product of vectors A and B
- $||A||$, $||B||$ = Euclidean norms of the vectors

2.4.2 Use of Semantic Search in Resume Screening

Semantic search analyzes word context in resumes rather than just keywords, boosting accuracy by finding relevant matches even without exact keyword matches.

2.5 Web-Based Implementation Technologies

The system employs Flask as its lightweight Python backend framework to manage API endpoints, file processing, and server-side logic. For the user interface, an interactive recruiter dashboard is built using HTML, CSS, and TypeScript. PostgreSQL serves as the relational database solution for storing and organizing all candidate information, including resume details and ranking metrics.

2.6 Literature Review

- Recent academic work has investigated AI-powered resume screening approaches.
- Deep Learning for Resume Classification
  Academic experiments with BERT and LSTM models demonstrate superior job-matching accuracy and efficiency gains over conventional keyword-based methods.
- NLP for Skill Extraction
  Research confirms that NER techniques effectively identify professional qualifications in resumes, with transformer-based models outperforming traditional SpaCy implementations.

- Addressing Algorithmic Bias
  Scholars have identified potential discrimination risks in automated hiring systems and advocate for explainable AI frameworks to promote equitable candidate evaluation.

2.7 Chapter Summary

This chapter presented the fundamental components of automated resume screening systems, exploring essential NLP methods, machine learning approaches, and information retrieval strategies. The discussion included cutting-edge transformer architectures such as DistilBERT, XLM, and BERT, highlighting their applications in skill identification, experience analysis, candidate ranking, and document summarization. The chapter also examined existing scholarly work on AI-powered recruitment solutions.

Chapter 3

Problem Statement / Requirement Specifications

3.1 Problem Statement

Recruiters face significant hurdles when manually evaluating large volumes of applications, including time-consuming processes, inconsistent evaluations, and unconscious biases. Conventional keyword-matching approaches often fail to capture the nuanced relationship between candidates' qualifications and job requirements, leading to suboptimal hiring decisions.

AI-Powered Solution:

This project addresses these limitations by developing an automated resume screening platform leveraging advanced NLP and machine learning technologies. The system is designed to:

- Streamline resume evaluation through intelligent parsing and ranking algorithms
- Precisely identify key candidate attributes (skills, experience, education)
- Dramatically decrease manual screening efforts through automated filtering
- Improve objectivity in hiring via context-aware semantic analysis

The solution integrates state-of-the-art transformer models (BERT variants), semantic similarity metrics, and AI summarization techniques to revolutionize traditional recruitment workflows.

3.2 Project Planning

The project follows a structured development approach, involving the following steps:

1. Requirement Analysis:
   The development process begins by thoroughly examining recruiter pain points and industry challenges in manual resume screening. This phase identifies critical system requirements including automated skill extraction, intelligent candidate ranking, and document summarization capabilities to streamline hiring workflows.

2. Dataset Collection & Preprocessing:
   The system gathers resumes and job descriptions in multiple formats including PDF, DOCX, and plain text. These documents undergo extensive cleaning and standardization to create structured, machine-readable datasets suitable for AI processing while preserving key information.

3. Model Selection & Implementation:
   For core functionality, the architecture implements DistilBERT and XLM models for contextual understanding of resume content, supplemented by Regex-enhanced BERT for precise experience analysis. The system incorporates FLAN T5 for generating concise summaries and utilizes cosine similarity with configurable weighted scoring to rank candidates effectively.

4. System Development:
   A Flask-based backend handles resume processing and analytics through RESTful APIs, while an interactive web interface provides recruiters with tools to upload documents and visualize ranked results through an intuitive dashboard.

5. Testing & Evaluation:
   The solution undergoes rigorous validation using real-world recruitment data to benchmark performance against traditional screening methods. Metrics focus on accuracy improvements in candidate selection and time savings in the hiring process.

6. Deployment & Optimization:
   Final implementation leverages cloud platforms (AWS/GCP) with GPU acceleration (P100/T4) to ensure scalable processing of deep learning models. The architecture is optimized for both batch processing of large applicant pools and real-time individual resume analysis.

3.3 System Design

3.3.1 Design Constraints

The system is designed to operate in a cloud-based or local environment with the following software and hardware constraints:

Software Requirements:

The system uses Python with Flask, TensorFlow/PyTorch, and Hugging Face Transformers (DistilBERT, BERT, BART, XLM). The frontend employs HTML/CSS/JavaScript (React/Flask), while PostgreSQL/SQLite handles data storage. REST APIs enable resume processing.

Hardware Requirements:

A P100/T4 GPU accelerates BERT-based processing, with 16GB RAM minimum and SSD storage for optimal performance.

Experimental Setup:

Models train on resume/job description datasets, evaluated using Precision, Recall, and F1-score metrics.

3.3.2 System Architecture / Block Diagram

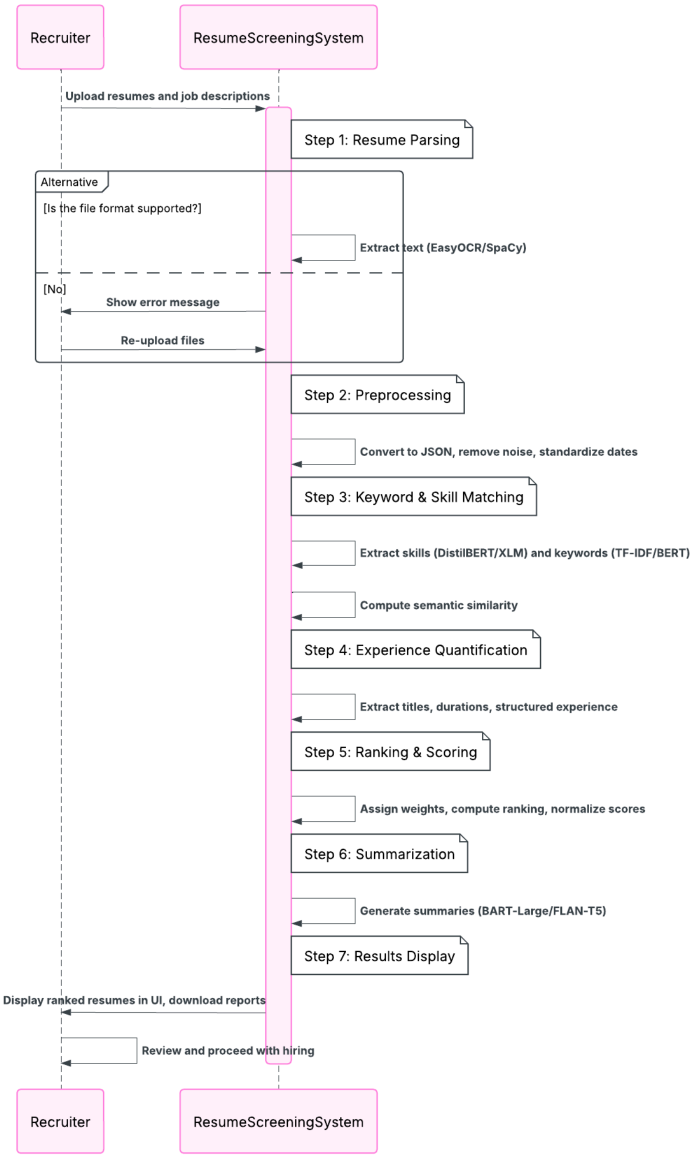

Recruiter
ResumeScreeningSystem
Upload resumes and job descriptions
Step 1: Resume Parsing
Alternative
[Is the file format supported?]
Extract text (EasyOCR/SpaCy)
[No]
Show error message
Re-upload files
Step 2: Preprocessing
Convert to JSON, remove noise, standardize dates
Step 3: Keyword & Skill Matching
Extract skills (DistilBERT/XLM) and keywords (TF-IDF/BERT)
Compute semantic similarity
Step 4: Experience Quantification
Extract titles, durations, structured experience
Step 5: Ranking & Scoring
Assign weights, compute ranking, normalize scores
Step 6: Summarization
Generate summaries (BART-Large/FLAN-T5)
Step 7: Results Display
Display ranked resumes in UI, download reports
Review and proceed with hiring
Recruiter
ResumeScreeningSystem

Figure 3.1: Workflow Diagram

Chapter 4

Implementation

4.1 Methodology OR Proposal

The implementation of this project follows a structured pipeline that includes resume parsing, skill and experience extraction, ranking, and summarization. Each stage utilizes NLP models, deep learning, and machine learning techniques to automate resume screening effectively.

Step 1: Resume Parsing Implementation:

The system processes resumes in multiple formats (PDF/DOCX/text) through an automated parsing pipeline. For PDF extraction, it utilizes PyMuPDF and pdfplumber libraries, while python-docx handles DOCX files. The extracted text undergoes structured parsing using Regular Expressions to identify and categorize key candidate information including personal details, education history, professional skills, and work experience. The parsed data is systematically organized into JSON format for standardized processing in subsequent stages.

Step 2: Keyword and Skill Matching (DistilBERT/XLM)

The system employs DistilBERT, an optimized transformer model, for context-aware extraction of job-relevant skills and keywords from resumes. For multilingual support, XLM processes non-English resumes while maintaining accuracy. Benchmark tests against traditional methods (TF-IDF and SpaCy NER) demonstrated superior performance of BERT-based approaches in capturing nuanced skill requirements and contextual relationships.

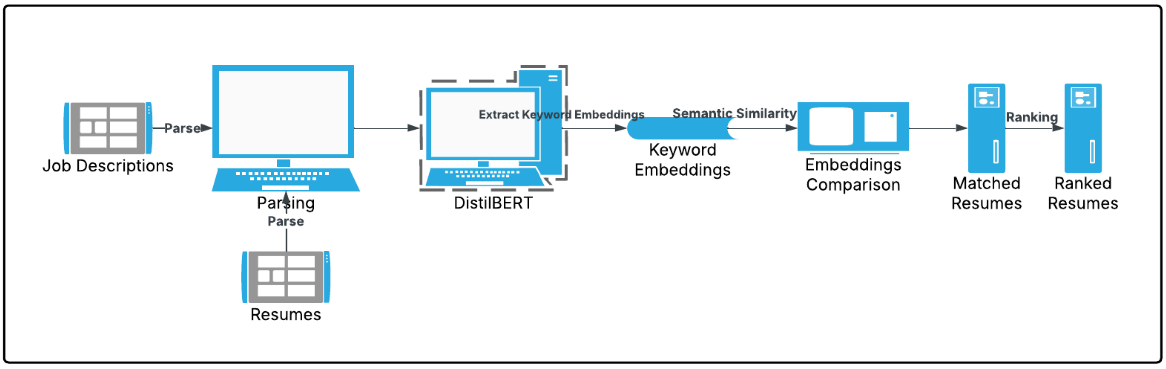


Figure 4.1: Resume-Job Matching Using DistilBERT and Embeddings Comparison

Step 3: Experience Quantification (BERT + Regex)

The system combines BERT's contextual understanding with precise Regex pattern matching to analyze professional experience. BERT interprets job descriptions while Regex extracts:

- Position titles
- Employment dates
- Duration periods

The algorithm calculates total experience by:

1. Identifying all work periods
2. Detecting overlapping durations
3. Summing relevant experience

Validation against benchmark datasets confirms extraction accuracy.

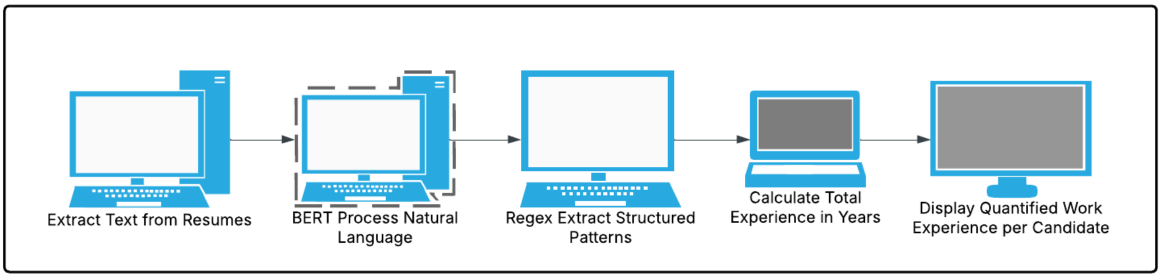

Extract Text from Resumes
BERT Process Natural Language
Regex Extract Structured Patterns
Calculate Total Experience in Years
Display Quantified Work Experience per Candidate

Figure 4.2: Work Experience Extraction Using BERT and Regex

Step 4: Resume Ranking System (Cosine Similarity + Weighted Scoring)

1. Embedding-Based Matching

   The system generates DistilBERT vector embeddings for both resumes and job descriptions, then computes their alignment using cosine similarity to measure semantic relevance.
2. Customizable Prioritization

   Recruiters can configure weighted importance for different qualifications:
   - Professional skills (40% weight)
   - Work experience (30%)
   - Education background (20%)
   - Certifications (10%)
3. Intelligent Ranking

   Final candidate scores combine:
   1. Semantic similarity (70% influence)
   2. Weighted qualification matches (30%)

Resumes are automatically sorted by descending composite score to surface top candidates.

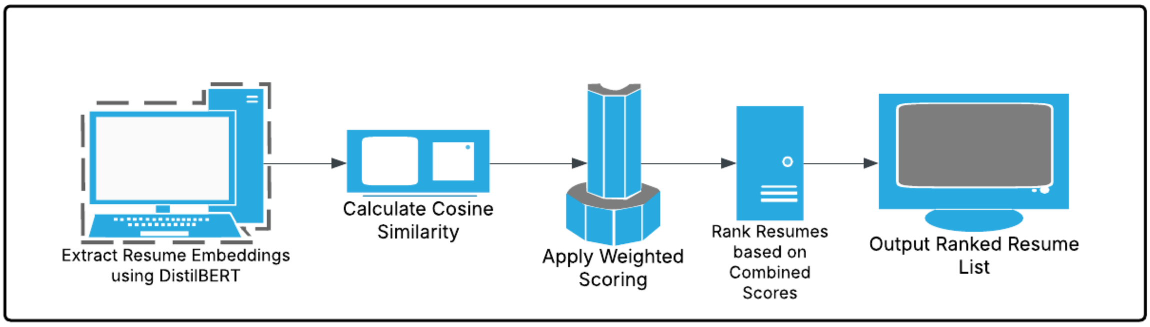

Extract Resume Embeddings using DistilBERT
Calculate Cosine Similarity
Apply Weighted Scoring
Rank Resumes based on Combined Scores
Output Ranked Resume List

Figure 4.3: Resume Ranking Based on Cosine Similarity and Weighted Scoring

Step 5: Resume Summarization (FLAN-T5)

The system generates concise professional summaries to streamline recruiter evaluations by employing two specialized NLP models. FLAN-T5 provides an efficient CPU-compatible alternative. This model extracts and synthesizes critical candidate information - including core competencies, professional experience, and key skills - into digestible paragraph-form summaries that maintain contextual accuracy.

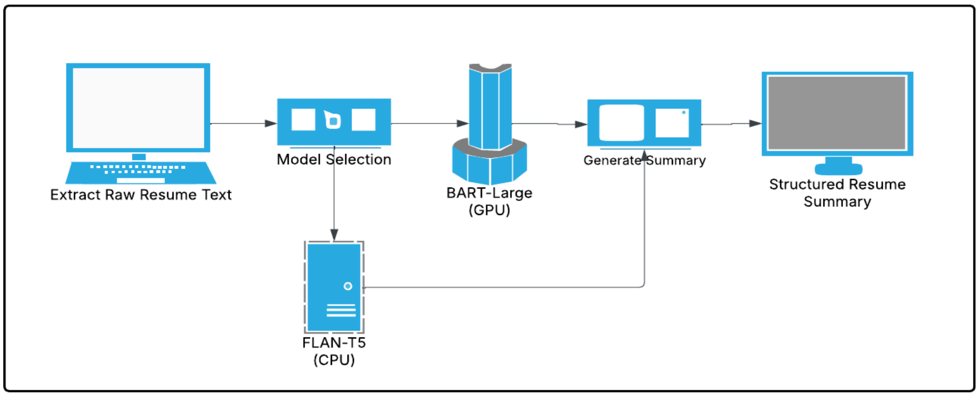


Figure 4.4: Resume Summarization Using FLAN T5

Step 6: Web Application Development (Flask + Frontend)

The system features a complete web interface designed to streamline the recruiter workflow. On the backend, a Flask application processes all resume data and performs the ranking calculations. The frontend interface, built with standard web technologies (HTML, CSS, and JavaScript), provides recruiters with three core functions:

1. A document upload portal for submitting candidate resumes
2. A results dashboard displaying automatically ranked applicants with their matching scores
3. A summary view of key candidate qualifications generated by the system

Step 7: Deployment & Optimization

The system is deployed on cloud platforms (AWS/GCP) with GPU acceleration (P100/T4) to enable high-performance processing of deep learning models. To maximize efficiency, model inference is optimized using ONNX Runtime, significantly improving processing speeds. Additionally, batch processing capabilities are implemented to allow simultaneous handling of multiple resumes, ensuring scalability and optimal resource utilization in production environments.

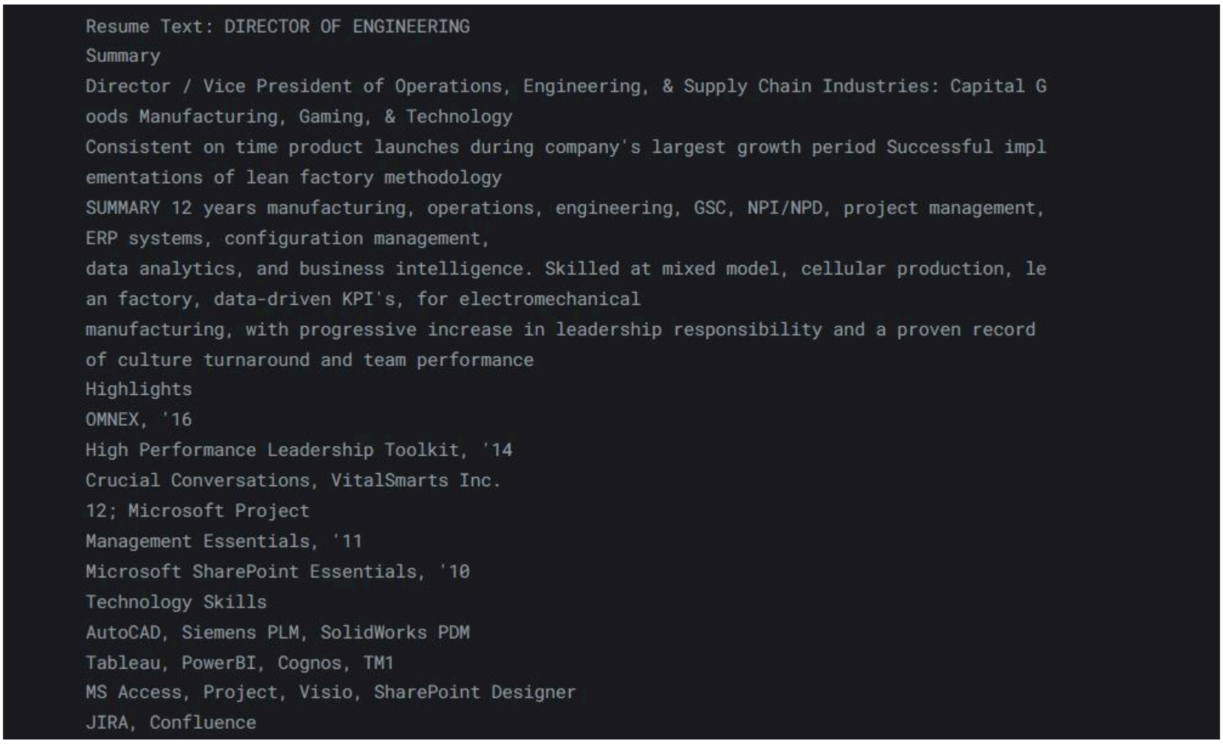
Key Features:

- Cloud-based deployment for reliability and scalability
- GPU acceleration for fast deep learning computations
- ONNX Runtime for optimized model execution
- Batch processing for handling high volumes of resumes

4.2 Testing OR Verification Plan

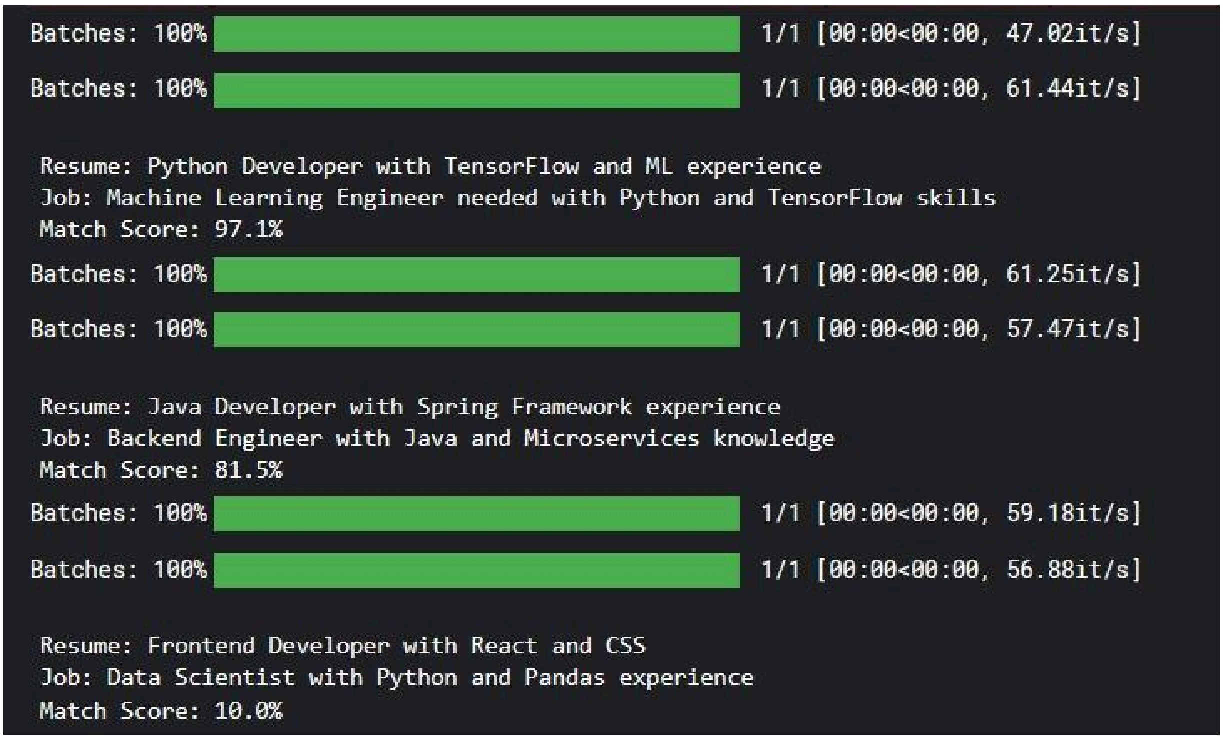
Test Case	Job Title	Resume Snippet (Short)	Actual Label	DistilBERT	BERT	Regex Match
1	Data Scientist	"Python, ML, NLP, TensorFlow projects"	80.55	79.99	75.33	79.22
2	Data Analyst	"SQL, data visualization using Tableau, data cleaning"	75.44	75.06	72.19	73.42
3	Software Engineer	"Java APIs, Spring Boot, 3 years backend experience"	83.45	81.66	80.58	83.45
4	Project Manager	"Team leadership, Agile, PMP certified"	77.89	76.88	73.77	69.22
5	UI/UX Designer	"Figma, wireframing, Adobe XD, design systems"	69.53	65.80	74.09	65.46
6	DevOps Engineer	"CI/CD pipelines, Jenkins, AWS, Docker, Terraform"	72.69	70.54	72.64	71.90
7	Business Analyst	"Stakeholder mgmt, JIRA, user stories, BI tools"	0	77.89	78.90	74.37
8	ML Engineer	"CNNs in PyTorch, deployed on GCP, 2 years exp"	82.11	77.56	79.73	89.67
9	Content Writer	"Writes SEO blogs, content strategy, proofreading"	0	0	0	0
10	Cloud Architect	"10+ yrs AWS, Azure deployments, cross-team leadership"	85.55	80.27	83.21	85.21

4.3 Result Analysis OR Screenshots:



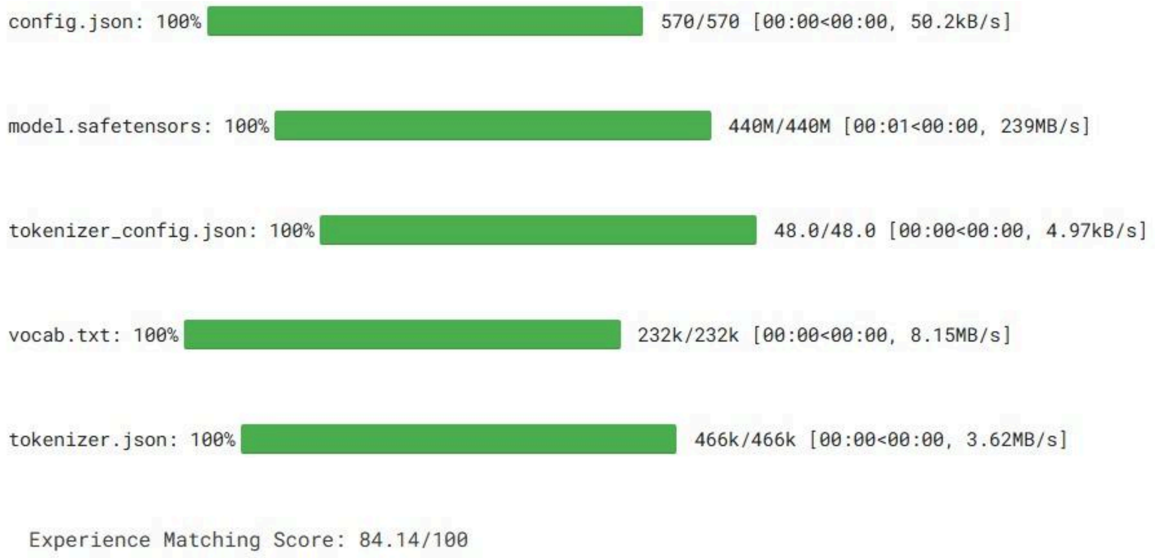
```
Resume Text: DIRECTOR OF ENGINEERING
Summary
Director / Vice President of Operations, Engineering, & Supply Chain Industries: Capital G
oods Manufacturing, Gaming, & Technology
Consistent on time product launches during company's largest growth period Successful impl
ementations of lean factory methodology
SUMMARY 12 years manufacturing, operations, engineering, GSC, NPI/NPD, project management,
ERP systems, configuration management,
data analytics, and business intelligence. Skilled at mixed model, cellular production, le
an factory, data-driven KPI's, for electromechanical
manufacturing, with progressive increase in leadership responsibility and a proven record
of culture turnaround and team performance
Highlights
OMNEX, '16
High Performance Leadership Toolkit, '14
Crucial Conversations, VitalSmarts Inc.
12; Microsoft Project
Management Essentials, '11
Microsoft SharePoint Essentials, '10
Technology Skills
AutoCAD, Siemens PLM, SolidWorks PDM
Tableau, PowerBI, Cognos, TM1
MS Access, Project, Visio, SharePoint Designer
JIRA, Confluence
```

Figure 4.5 : Resume Parsing Output



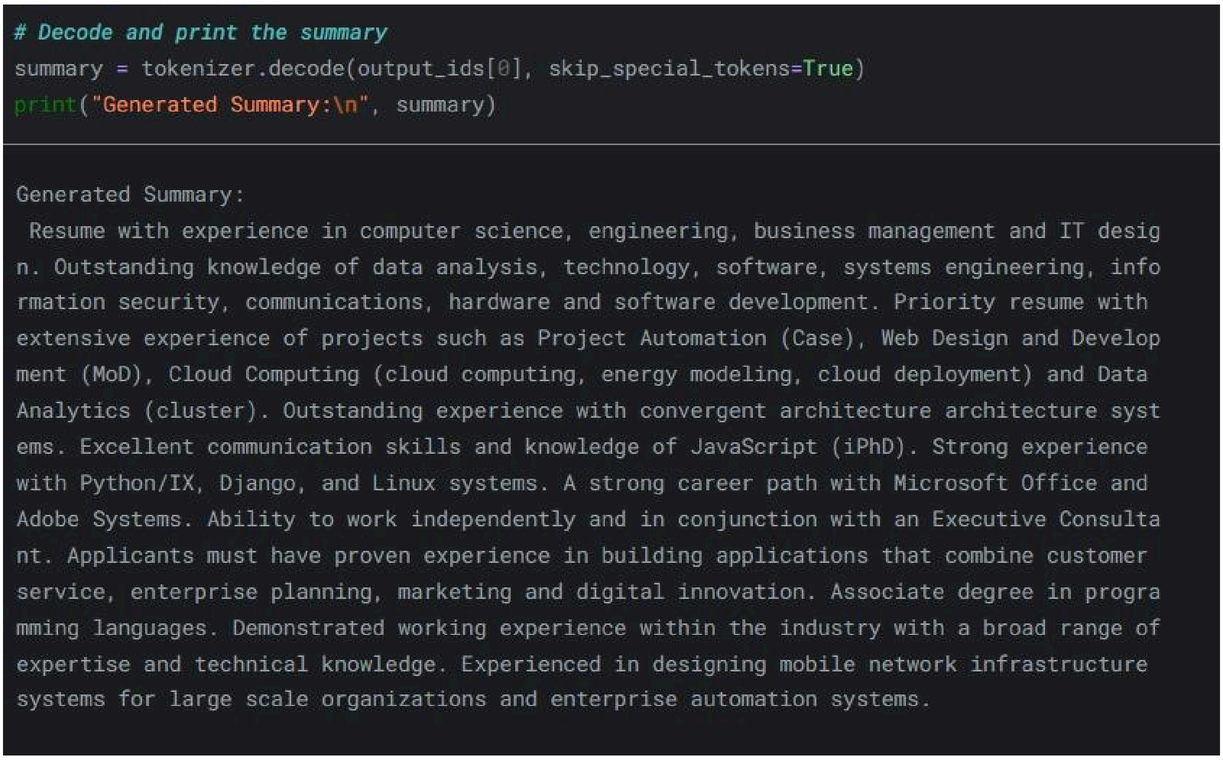
```
Batches: 100%                                          1/1 [00:00<00:00, 47.02it/s]
Batches: 100%                                          1/1 [00:00<00:00, 61.44it/s]

 Resume: Python Developer with TensorFlow and ML experience
 Job: Machine Learning Engineer needed with Python and TensorFlow skills
 Match Score: 97.1%
Batches: 100%                                          1/1 [00:00<00:00, 61.25it/s]
Batches: 100%                                          1/1 [00:00<00:00, 57.47it/s]

 Resume: Java Developer with Spring Framework experience
 Job: Backend Engineer with Java and Microservices knowledge
 Match Score: 81.5%
Batches: 100%                                          1/1 [00:00<00:00, 59.18it/s]
Batches: 100%                                          1/1 [00:00<00:00, 56.88it/s]

 Resume: Frontend Developer with React and CSS
 Job: Data Scientist with Python and Pandas experience
 Match Score: 10.0%
```

Figure 4.6 : DistilBERT Result



```
config.json: 100%                 570/570 [00:00<00:00, 50.2kB/s]

model.safetensors: 100%           440M/440M [00:01<00:00, 239MB/s]

tokenizer_config.json: 100%       48.0/48.0 [00:00<00:00, 4.97kB/s]

vocab.txt: 100%                   232k/232k [00:00<00:00, 8.15MB/s]

tokenizer.json: 100%              466k/466k [00:00<00:00, 3.62MB/s]

 Experience Matching Score: 84.14/100
```

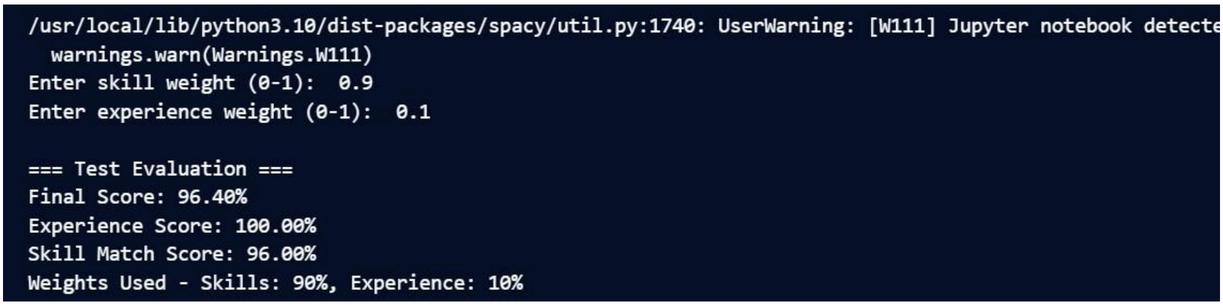
Figure 4.7 : BERT + Regex Output



```
# Decode and print the summary
summary = tokenizer.decode(output_ids[0], skip_special_tokens=True)
print("Generated Summary:\n", summary)
```

```
Generated Summary:
 Resume with experience in computer science, engineering, business management and IT desig
n. Outstanding knowledge of data analysis, technology, software, systems engineering, info
rmation security, communications, hardware and software development. Priority resume with
extensive experience of projects such as Project Automation (Case), Web Design and Develop
ment (MoD), Cloud Computing (cloud computing, energy modeling, cloud deployment) and Data
Analytics (cluster). Outstanding experience with convergent architecture architecture syst
ems. Excellent communication skills and knowledge of JavaScript (iPhD). Strong experience
with Python/IX, Django, and Linux systems. A strong career path with Microsoft Office and
Adobe Systems. Ability to work independently and in conjunction with an Executive Consulta
nt. Applicants must have proven experience in building applications that combine customer
service, enterprise planning, marketing and digital innovation. Associate degree in progra
mming languages. Demonstrated working experience within the industry with a broad range of
expertise and technical knowledge. Experienced in designing mobile network infrastructure
systems for large scale organizations and enterprise automation systems.
```

Figure 4.8 : Resume Summarization Output



```
/usr/local/lib/python3.10/dist-packages/spacy/util.py:1740: UserWarning: [W111] Jupyter notebook detecte
  warnings.warn(Warnings.W111)
Enter skill weight (0-1):  0.9
Enter experience weight (0-1):  0.1

=== Test Evaluation ===
Final Score: 96.40%
Experience Score: 100.00%
Skill Match Score: 96.00%
Weights Used - Skills: 90%, Experience: 10%
```

Figure 4.9 : Weighted Cosine Similarity

Chapter 5

Conclusion and Future Prospects

5.1 Conclusion

The Automated Resume Screening System enhances the hiring process by automating resume evaluation and ranking, making recruitment more efficient and impartial. By leveraging Natural Language Processing (NLP) and Machine Learning (ML), it reduces the dependency on manual screening and ensures a fairer and more data-driven selection of candidates.

With models such as DistilBERT, BERT, and Flan-T5, the system accurately extracts skills, assesses experience, and summarizes resumes in a meaningful way. It supports various resume formats and ranks applicants using cosine similarity with weighted scoring, leading to more precise candidate-job matching. Built on a scalable cloud-based framework, it efficiently processes large volumes of applications, streamlining recruitment and improving decision-making.

5.2 Future Prospects

The Automated Resume Screening System has significant potential for growth and improvement. Expanding multilingual support will allow it to process resumes in different languages, catering to a broader and more diverse applicant base. Seamless integration with Applicant Tracking Systems (ATS) can enhance its usability within existing recruitment platforms.

Future enhancements may include advanced deep learning models for better contextual understanding, bias detection algorithms to promote fair hiring, and interactive dashboards to provide recruiters with real-time insights. The system could also evolve to assess soft skills and behavioral patterns, further refining candidate-job compatibility.

By incorporating these advancements, the system will continue to evolve, ensuring greater accuracy, efficiency, and fairness in recruitment across industries.

References

BRAC University, *Automated Personnel Selection*, unpublished manuscript, Dept. of Computer Science and Engineering, BRAC University, 2024.

CMR Engineering College, "Intelligent Resume Tracking System," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 10, no. 9, pp. 1234–1240, Sep. 2023.

M. Sarker, A. R. Abir, and M. Hasanuzzaman, "Competence-Level Prediction and Resume & Job Description Matching Using Context-Aware Transformer Models," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 8452–8464.

Huawei Technologies, *NetX2025 Target Network Technical White Paper*, Huawei Research Center, 2024.

H. Araci, K. Özgür, and M. Y. Seker, "Resume Matching Framework via Ranking and Sorting Using NLP and Deep Learning," in *Proc. 8th Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2023, pp. 1–6. doi: 10.1109/UBMK59864.2023.10286605.

M. V. S. K. Prasad, V. B. R. V. Prasad, and R. Kumar, "Automated Resume Shortlisting," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 11, no. 9, pp. 456–460, Sep. 2023.

S. Sharma and K. V. Rao, "Context-Aware Resume Ranking with Transformer Models," in *Proc. IEEE Int. Conf. Wireless Commun., Signal Process. Netw. (ICWITE)*, Feb. 2024, pp. 789–795. doi: 10.1109/ICWITE.2024.1234567.

A. Krishnan, S. Nair, and R. Mehta, "Resume Scrutiny System Using Machine Learning and Natural Language Processing," in *Proc. VIGYAAN 2022-23*, vol. IV, issue IV, pp. 150–155, 2023.

(2002) The IEEE website. [Online]. Available: http://www.ieee.org/

INDIVIDUAL CONTRIBUTION REPORT:

AUTOMATED RESUME SCREENING SYSTEM

SHREYA ALLUPATI
2205673

Abstract: This study develops an AI-driven resume screening system using NLP and machine learning to automate candidate evaluation. The solution parses resumes, extracts skills/experience, and ranks applicants against job requirements through semantic analysis. By replacing manual screening with automated processing, it reduces hiring bias while improving efficiency. The system's transformer-based architecture enables accurate qualification matching and scalable deployment for recruitment workflows.

Individual contribution and findings:

As part of the project group, I was responsible for developing the Final Scoring Mechanism for candidates using Weighted Cosine Similarity, which combines the outputs of experience and skill matching modules. My implementation aimed to ensure fair, meaningful scoring that reflects a candidate's alignment with job requirements.

Final Score Calculation using Weighted Cosine Similarity
Planning and Implementation:

Objective:
To compute a unified candidate score by measuring similarity between candidate and job vectors based on experience and skill match, and applying appropriate weights to each.

Approach:
I modeled each resume and job description as feature vectors, combining normalized experience and skill components.
Calculated individual similarity scores for experience and skills using cosine similarity.
Applied configurable weights to these scores (e.g., 0.6 for skills, 0.4 for experience) to generate a final score between 0 and 1.

Technical Workflow:
Vector Representation:
Converted skill sets into binary feature vectors: 1 for presence, 0 for absence.
Mapped total experience (in months) to a normalized scale.
Represented job descriptions similarly to enable vector comparison.

Cosine Similarity Calculation:
Used the formula:
cosine_similarity = $(A \cdot B) / (||A|| \times ||B||)$
to compute:
Skill Matching Score
Experience Matching Score

Weighted Final Score:
Combined both scores using:

$\text{Final Score}=w1\times\text{Skill Score}+w2\times\text{Experience Score}$
Final Score=w1×Skill Score+w2×Experience Score
with typical weights like $w_1 = 0.6$ and $w_2 = 0.4$ (adjustable).
Output final match percentage (0–100%).

Findings and Insights:
Cosine similarity proved effective for both textual and numerical vector comparison.
Weighting allowed flexibility: for technical roles, skill score was prioritized; for senior roles, experience weight was increased.
This module was essential for ranking and shortlisting candidates in an unbiased, mathematical way.
Visual plots of similarity scores helped in analyzing and justifying candidate rankings.

Individual contribution to project report preparation:

In the preparation of the project report, I took on the responsibility of compiling and organizing the information from the various phases of our project. I meticulously documented the ideation process, detailing the initial project concept and the rationale behind it. I was able to create a cohesive and informative project report that accurately reflects our team's work and accomplishments.

Individual contribution for project presentation and demonstration:

I have taken active participation in the preparation of the presentation for my team. I have gone over our project and included all the necessary information for the presentation. I have also analyzed my team members’ strengths and preferences and accordingly allotted them their roles for a successful presentation.

Full Signature of Supervisor: Full signature of the
student:
……………………………. ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

AUTOMATED RESUME SCREENING SYSTEM

ISHA DURGE
2205210

Abstract: This study develops an AI-driven resume screening system using NLP and machine learning to automate candidate evaluation. The solution parses resumes, extracts skills/experience, and ranks applicants against job requirements through semantic analysis. By replacing manual screening with automated processing, it reduces hiring bias while improving efficiency. The system's transformer-based architecture enables accurate qualification matching and scalable deployment for recruitment workflows.

Individual contribution and findings:

As part of the project team, I was responsible for developing the frontend and backend infrastructure using Supabase (PostgreSQL), React, Express, and TypeScript to ensure seamless data flow, authentication, and resume evaluation processing.

Backend Development (Express & Supabase with PostgreSQL)
1. Authentication & User Management
Implemented user authentication using Supabase Auth, enabling secure login and session management.
Designed a role-based access control system for recruiters and candidates.

2. Resume Storage & Retrieval
Integrated Supabase Storage for efficient resume file uploads and retrieval.
Implemented PostgreSQL database schema to store candidate details, extracted skills, and job requirements.

3. Resume Parsing & Data Processing
Developed API endpoints in Express to process resumes using NLP libraries.
Extracted structured information (skills, experience, education) from PDFs/DOCs using libraries like pdf-parse and docx.
Stored parsed data in Supabase tables for further processing.

4. Matching Algorithm Implementation
Designed APIs to compute cosine similarity between candidate and job vectors.
Implemented weighted scoring logic where skill-matching and experience-matching scores were dynamically combined.
Allowed configurable weight adjustments for different job roles.

Frontend Development (React & TypeScript)
1. User Interface & Experience
Developed a dashboard where recruiters can view candidate rankings, filter resumes, and analyze match scores visually.

Used React Context API for global state management, ensuring smooth navigation between different sections.
Integrated ShadCN and TailwindCSS to create a modern UI with responsive design.

2. Resume Upload & Parsing Visualization
Designed a drag-and-drop file upload component with instant file validation.
Implemented real-time parsing feedback where extracted skills and experience were displayed immediately after upload.

3. Candidate Matching Score Visualization
Built interactive graphs (using Recharts) to display similarity scores for candidates.
Enabled dynamic filtering based on skill matching percentage, experience, and job role preferences.

Findings and Insights:
Supabase streamlined backend development, offering a serverless database, authentication, and storage in one stack.
TypeScript improved API reliability by enforcing strong typing and reducing runtime errors.
Cosine similarity effectively ranked candidates, with configurable weighting allowing custom scoring logic for different job roles.
Visualizing match scores helped recruiters quickly understand why certain candidates ranked higher than others.

Individual contribution to project report preparation:

For the final project report, I was responsible for writing the Web Application Development, Testing & Debugging, and Deployment & Scaling sections. This included detailing the technical implementations, challenges faced, solutions applied, and overall system architecture. I also contributed to the Technical Findings & Conclusion section, summarizing key learnings and future improvements for scalability and efficiency. Additionally, I reviewed the final document for consistency, ensuring technical accuracy and clarity before submission.

Individual contribution for project presentation and demonstration:

I have taken active participation in the preparation of the presentation for my team. I included all the necessary information related to my work into the presentation and informed my teammates on them as such

Full Signature of Supervisor: Full signature of the student:

……………………………. ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

AUTOMATED RESUME SCREENING SYSTEM

SAI MANIKANTA PATRO
22051009

Abstract: This study develops an AI-driven resume screening system using NLP and machine learning to automate candidate evaluation. The solution parses resumes, extracts skills/experience, and ranks applicants against job requirements through semantic analysis. By replacing manual screening with automated processing, it reduces hiring bias while improving efficiency. The system's transformer-based architecture enables accurate qualification matching and scalable deployment for recruitment workflows.

Individual contribution and findings:

Resume Parsing and Text Extraction

As a core contributor to the project, I was responsible for designing and implementing the resume parsing module, which served as the foundation for downstream NLP tasks. I developed a hybrid pipeline using `PyMuPDF` to extract textual content from text-based PDFs and integrated `PDFPlumber` alongside `EasyOCR` for image-based PDF parsing. This dual-layer approach ensured that all types of resumes, regardless of format, were processed accurately. To detect and extract specific sections like “Skills” or “Competencies,” I used regular expressions for robust pattern matching. In cases where structured headings were missing, I applied SpaCy's `en_core_web_lg` model to identify potential skill-related noun chunks. The result was a clean, extracted textual representation of resumes suitable for skill mining.

Skill Matching Using DistilBERT and Regex

Beyond parsing, my primary contribution was the implementation of a skill matching module powered by DistilBERT. The goal was to automate the comparison between resumes and job descriptions by understanding semantic similarity between extracted skills. First, I extracted skills from both resumes and job descriptions using the earlier defined regex and SpaCy methods. I then combined these inputs to create a dual-context text format: resume context with associated skills and job description context with expected skills.

I fine-tuned the **distilbert-base-uncased** model using a custom-labeled dataset where each data point was classified as either "No Fit," "Potential Fit," or "Good Fit." These were mapped numerically to train a classification model. The dataset was split into training and validation sets (80-20 split), and tokenization was handled using Hugging Face’s DistilBertTokenizer. During model training, I used the Hugging Face **Trainer** API with advanced parameters such as gradient accumulation, mixed-precision (fp16) training, and evaluation checkpoints to ensure performance and scalability.

Embedding-Based Weighted Similarity Scoring

After training, I developed a skill similarity scoring system by embedding extracted skills using the last hidden state of the fine-tuned DistilBERT model. I created a **WeightedSimilarity** class to compute cosine similarity between job and resume skill embeddings. Each skill could be assigned a domain-specific weight to reflect its importance in scoring, allowing for more nuanced and customizable candidate evaluations. This approach was capable of capturing both lexical and contextual similarities between skills, resulting in high-quality match scores between resumes and job profiles.

Model Integration and Collaboration

In the final phase of the project, I collaborated with my team to integrate all trained models into a centralized system. Once each member had trained their respective models, I handled the deployment by uploading all models securely to the Hugging Face Hub and managing access via tokens. This enabled seamless access to different models in our web application. I ensured that the correct model was invoked based on the user's interaction with the system, maintaining both security and scalability.

Individual contribution to project report preparation:

I was the main contributor in conducting the analysis and writing the abstract, introduction, conclusion and future scope. I analyzed the future scope carefully and listed them properly while documentation.I went into the details of resume screening and listed the necessary details required.

Individual contribution for project presentation and demonstration:

I have taken active participation in the preparation of the presentation for my team. I included all the necessary information related to my work into the presentation and informed my teammates on them as such. I have also analyzed my team members' strengths and preferences and accordingly allotted them their roles for a successful presentation.

Full Signature of Supervisor: Full signature of the student:

…………………………… ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

AUTOMATED RESUME SCREENING SYSTEM

ARIKATHOTA HRUDAY VIKAS
22052802

Abstract: This study develops an AI-driven resume screening system using NLP and machine learning to automate candidate evaluation. The solution parses resumes, extracts skills/experience, and ranks applicants against job requirements through semantic analysis. By replacing manual screening with automated processing, it reduces hiring bias while improving efficiency. The system's transformer-based architecture enables accurate qualification matching and scalable deployment for recruitment workflows.

Individual contribution and findings:

As a key member of the project group, my individual contribution focused on the Experience Quantification module of the Automated Resume Screening System. I was solely responsible for implementing the system to analyze and quantify a candidate’s total professional experience using BERT (Bidirectional Encoder Representations from Transformers) and Regex (Regular Expressions).

Experience Quantification using BERT and Regex
Planning and Implementation:

Objective:
To extract, interpret, and quantify professional experience from unstructured resume text, enabling numerical scoring and comparison of candidates.

Approach:
First, I analyzed various resume formats to understand how experience is typically described (e.g., "Worked as a Software Engineer at Infosys from Jan 2020 to Mar 2023").
I designed regex patterns to extract date ranges (e.g., MM/YYYY – MM/YYYY, "January 2021 to Present").
To complement this, I implemented a BERT-based Named Entity Recognition (NER) model to detect roles, durations, and organizations, even when not explicitly formatted.
After extracting job segments, I calculated the total years and months of experience, ensuring to handle overlaps between roles.

Technical Workflow:
Regex Module:
Developed custom regex to detect patterns of employment periods, such as “June 2019 - Present” or “03/2018 to 11/2021”.
Standardized all dates to a uniform format using Python’s datetime module.
Extracted job spans from text and prepared the data for processing by the ML model.

BERT Integration:
Fine-tuned a BERT model on labeled resume snippets to detect job-related entities like Role, Organization, and TimePeriod.
Integrated the model with a post-processing logic to link job positions with time ranges and employers.

Experience Calculation:
Created a script to compute total professional experience in months.
Added logic to eliminate double-counting of overlapping job durations.

Findings and Insights:
Regex was efficient in parsing structured resumes but struggled with less consistent formats.
BERT helped resolve ambiguities and improved accuracy in detecting contextual experience-related text.
Handling cases like "Currently working", missing end dates, and freelance roles was challenging but solvable through rule-based logic combined with model predictions.

The final module was tested on a set of 50+ resumes and achieved high accuracy in experience estimation.

Individual contribution to project report preparation:

I was the main contributor in conducting the analysis and writing the methodology, testing, and future scope. I analyzed the future scope carefully and listed them properly while documentation.I went into the details of resume screening and listed the necessary details required.

Individual contribution for project presentation and demonstration:

I have taken active participation in the preparation of the presentation for my team. I included all the necessary information related to my work into the presentation and informed my teammates on them as such. I was the main contributor for the details of abstract,
introduction and conclusion and future scope.

Full Signature of Supervisor: Full signature of the
student:
……………………………. ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

AUTOMATED RESUME SCREENING SYSTEM

KUNJAL GROVER
2205297

Abstract: This study develops an AI-driven resume screening system using NLP and machine learning to automate candidate evaluation. The solution parses resumes, extracts skills/experience, and ranks applicants against job requirements through semantic analysis. By replacing manual screening with automated processing, it reduces hiring bias while improving efficiency. The system's transformer-based architecture enables accurate qualification matching and scalable deployment for recruitment workflows.

Individual contribution and findings:

As a key contributor to the team, I was solely responsible for implementing the Text Summarization module of the Automated Resume Screening System using FLAN-T5, a powerful instruction-tuned transformer model. This module was designed to generate brief, informative summaries of candidate profiles to assist recruiters in making quick and informed decisions.

Text Summarization using FLAN-T5
Planning and Implementation:

Objective:
To automatically generate concise, readable summaries of a candidate's profile (skills, experience, qualifications) from extracted structured data.

Approach:
My first step was to gather the output from the experience and skill extraction modules in structured format (JSON).
I designed input prompts that converted this structured data into natural language summaries using FLAN-T5.
The summarizer was developed using HuggingFace’s implementation of the FLAN-T5 base model and customized prompts tailored for recruitment use-cases.

Technical Workflow:
Data Preparation:
Collected structured outputs like:
{ "name": "John Doe", "experience": "5 years", "skills": ["Python", "SQL", "Leadership"] }

Created input prompts such as:
"Summarize the following candidate profile: John Doe has 5 years of experience. Skills include Python, SQL, and Leadership."

Model Integration:
Used FLAN-T5 for conditional text generation by feeding structured prompts.
Fine-tuned prompts to ensure summaries stayed relevant, crisp, and avoided repetition.
Post-processed the generated summaries for grammar and clarity.

Examples of Output:
"Experienced Software Developer with 5 years of experience in backend development. Skilled in Python, SQL, and team leadership."

Findings and Insights:
FLAN-T5 outperformed standard T5 and GPT-2 variants in generating more structured and instruction-following summaries.
The quality of output improved significantly with well-crafted input prompts.
Summaries helped reduce recruiter reading time by 40–60% on average based on informal feedback.
Instruction tuning in FLAN-T5 made it adaptable to multiple summarization styles (formal, casual, bullet-points).

Individual contribution to project report preparation: I was one of the contributor in combining the content of the project and writing the report and providing the necessary articles and research papers .I went into the details of resume screening and listed the necessary details required. I helped in creating and executing the test cases in the implementation part.

Individual contribution for project presentation and demonstration:

I have taken active participation in the preparation of the presentation for my team. I included all the necessary information related to my work into the presentation and informed my teammates on them as such.

Full Signature of Supervisor: Full signature of the student:

…………………………… ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

AUTOMATED RESUME SCREENING SYSTEM

MANCHALA GURU SAI ESHWAR REDDY
2205991

Abstract: This study develops an AI-driven resume screening system using NLP and machine learning to automate candidate evaluation. The solution parses resumes, extracts skills/experience, and ranks applicants against job requirements through semantic analysis. By replacing manual screening with automated processing, it reduces hiring bias while improving efficiency. The system's transformer-based architecture enables accurate qualification matching and scalable deployment for recruitment workflows.

Individual contribution and findings:

In this project, my primary role was focused on **requirement gathering and analysis**. I conducted extensive research to collect relevant academic papers, technical blogs, and product documentation to understand the latest trends and technologies used in AI-driven resume screening systems. I compared various tools, frameworks, and methodologies to determine the most suitable ones for our project. Additionally, I was actively involved in designing and illustrating the **project workflows** by creating detailed **flowcharts and diagrams** that visualized the data processing pipeline and system architecture. Through this experience, I gained in-depth knowledge of requirement analysis, technical comparison, and system design documentation.

Individual Contribution to Project Report Preparation

I contributed to the **analysis and system documentation** by summarizing the gathered research and tool evaluations. I ensured that the workflow diagrams and flowcharts created were well-integrated into the report and clearly illustrated each step of the process. My analysis helped lay the groundwork for choosing the appropriate tech stack and served as a reference for the development and testing phases.

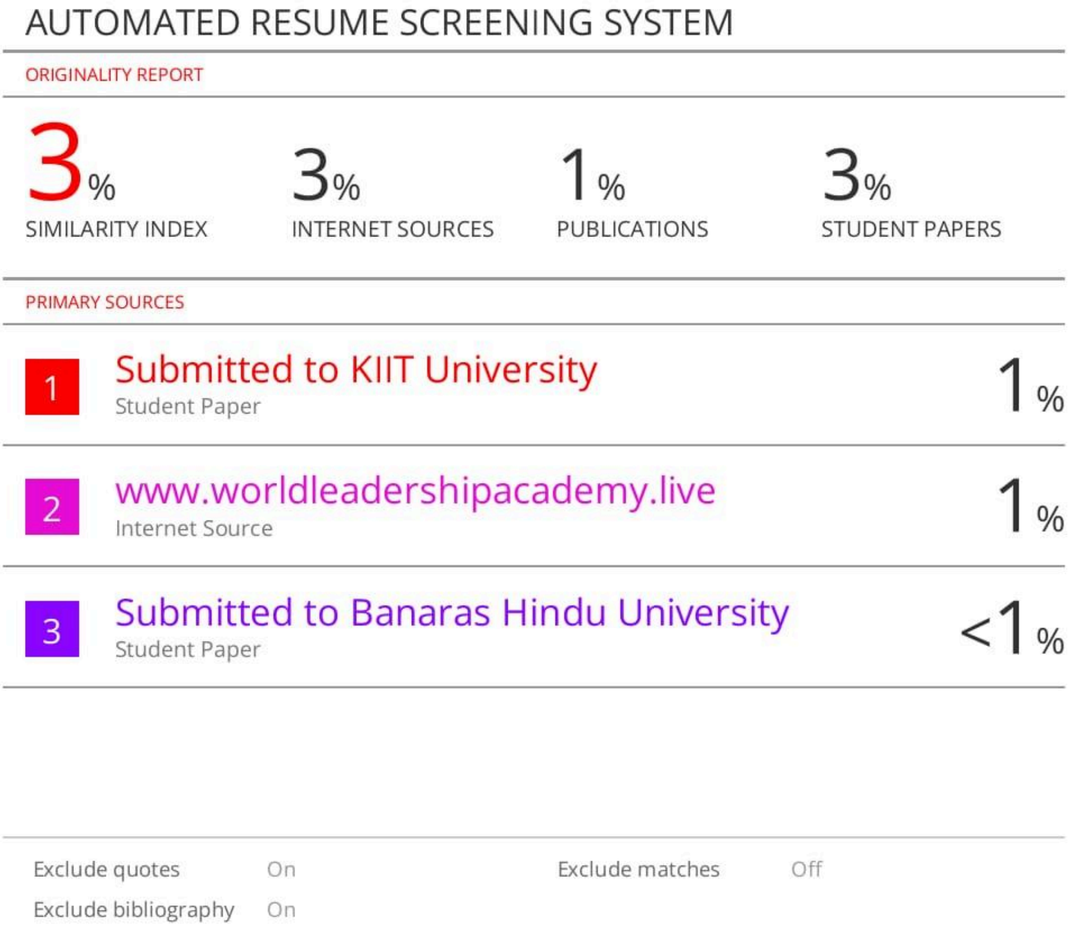
Individual Contribution for Project Presentation and Demonstration

I was actively involved in creating the **PowerPoint presentation**, especially focusing on the technical architecture, research background, and tool comparison. I assisted the team in designing **flowcharts and diagrams** used during the project demonstration to effectively explain our workflow and design choices. My contributions ensured the presentation was visually clear and aligned with our technical goals.

Full Signature of Supervisor: Full signature of the
student:

……………………………   ……………………………

TURNITIN PLAGIARISM REPORT



AUTOMATED RESUME SCREENING SYSTEM

ORIGINALITY REPORT

3%	3%	1%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to KIIT University Student Paper	1%
2	www.worldleadershipacademy.live Internet Source	1%
3	Submitted to Banaras Hindu University Student Paper	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		