IJO -INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND ENGINEERING

(ISSN: 2814-1881) ADITYA JAGTAP *

https://ijojournals.com/ *Volume 08 || Issue 04 || April, 2025 ||*

"DOCUMENT-DRIVEN AI SOLUTIONS FOR TALENT OPTIMIZATION"

DOCUMENT-DRIVEN AI SOLUTIONS FOR TALENT OPTIMIZATION

Team Members: Aditya Jagtap, Saloni Mahadik, Sahil Utekar, Priyal Kalal,
Guide: Dr Pankaj Agarkar,
Department of Computer Enginnering
Ajeenkya DY Patil School of Engineering, Pune

***Abstract*—**The recruitment process is a critical function in modern organizations, and manual screening of resumes is often time-consuming and error-prone. This paper proposes a structured and semi-automated framework for extracting key resume attributes using AI and ranking candidates through a configurable scoring model. By integrating natural language processing, automation tools, and visualization platforms, the system enhances decision-making efficiency for recruiters. The solution uses modular Python scripts and APIs to parse resumes, score candidates, and present the results through dashboards, with an optional feedback loop for further evaluation. In particular, the proposed system leverages the capabilities of GPT-3.5 through Lang Chain to semantically understand resume content, enabling more accurate parsing and ranking. The system architecture supports end-to-end automation, starting from resume intake via email, intelligent parsing, rule-based scoring, and finally, interactive result visualization.

Keywords*—Recruitment, Resume parsing, Candidate ranking, Recruitment automation, GPT, Lang Chain, Power BI, NLP*

I. INTRODUCTION

Efficient Recruitment has undergone significant transformation in recent years with the adoption of technology. Despite these advancements, resume screening remains a labor-intensive task, particularly when faced with large applicant pools. Manual evaluation not only consumes considerable time but is also susceptible to human bias and oversight. As organizations seek data-driven hiring methods, there is a growing interest in intelligent systems that can assist in resume evaluation, candidate ranking, and recruitment communication.

Traditional applicant tracking systems often fail to interpret complex resume structures and struggle with information hidden in unstructured formats. This leads to important skills or experiences being overlooked, ultimately affecting hiring quality. With the emergence of AI and large language models, it is now feasible to automate parts of the hiring pipeline that once required extensive human effort.

This paper presents a project designed to address these challenges using AI-based tools. The goal is to automate resume intake, parse and extract key details using natural language processing, score and rank candidates against configurable criteria, and present the results using dashboards. The system aims to reduce manual workload while increasing the objectivity and efficiency of recruitment workflows. This approach is modular, scalable, and allows real-time insights for HR professionals and decision-makers.

II. CONCEPTS AND TECHNOLOGIES USED

In this section, we define key concepts and terminologies while explaining the underlying technologies that form the foundation of our work.

Resume screening is an essential process in human resource management, where candidate profiles are evaluated against job requirements to shortlist potential hires. Traditionally, this process is manual, time-consuming, and prone to errors. Automating resume screening and job matching enhances recruitment efficiency by reducing time spent on manual tasks and ensuring alignment between candidate skills and job roles.

A. Natural language processing in resume screening

NLP techniques enable the extraction of structured information from unstructured text such as resumes. These techniques are used to identify candidate qualifications, work experience, technical skills, and certifications. By transforming text data into structured formats, NLP allows automated comparison and scoring based on predefined criteria.

The project relies on AI language models, which are capable of understanding contextual relationships in language. These models are useful for extracting nuances in professional profiles and evaluating experience relevance with respect to job requirements.

B. GPT-3.5 and Lang Chain Framework

GPT-3.5 is a large language model developed by Open AI, capable of understanding and generating human-like text. It can interpret resumes holistically, identifying various sections and extracting semantic meaning from each. Lang Chain serves as an orchestration framework, enabling prompt design, chaining logic, and parsing outputs into usable structures like JSON or CSV.

Lang Chain also simplifies the handling of token limits by allowing long resumes to be split and parsed incrementally. The combination of GPT and Lang Chain enables intelligent and consistent extraction of resume data.

C. Email Intake and Automation

The Gmail API is integrated into the system to automate the intake of resumes sent via email. In case of limited API access, Robotic Process Automation (RPA) scripts are used to simulate email download and attachment extraction. This ensures a continuous and automated intake pipeline without manual intervention.

D. Dashboard Visualization

Power BI is utilized to present candidate information in a user-friendly, interactive dashboard. Recruiters can filter candidates by score, skill match, or experience level. Interviewers can use dashboards to prepare for evaluation, and business leads can track hiring trends and pipeline performance.

E. Configuration-Driven Scoring

A JSON configuration file stores the scoring weights and criteria, allowing for dynamic control over how candidates are evaluated. Parameters such as skill match, project relevance, certifications, and cultural fit can be adjusted without modifying the core logic. This enhances the system's adaptability to different hiring needs.

III. METHODOLOGY

The The primary objective of this research is to develop an AI-powered framework for automating resume screening and streamlining the job-matching process. This system aims to optimize talent acquisition and resource management by leveraging advanced technologies such as Open AI's GPT API, PyPDF2, and frameworks like Lang Chain..

- A. API Setup and Authentication

The system integrates with Open AI's GPT-3.5 API through the Lang Chain framework to facilitate semantic resume parsing and job matching. Secure API access is established using token-based authentication. Lang Chain handles the orchestration of prompts, error management, and result structuring, enabling reliable interaction with the LLM. This setup supports the parsing of resumes in a structured, modular manner and prepares the foundation for scoring and ranking workflows.

- B. Data Retrieval

Resume data is primarily collected through two automated methods:

1. Gmail API Integration: Resumes are received as attachments via a dedicated recruitment email address. Python scripts interact with the Gmail API to download and log resume files.
2. Manual JD Repository (Optional): Job descriptions are managed via a standardized .txt format stored in designated folders, enabling manual or batch upload for job matching purposes.

This approach ensures scalable and streamlined intake of resumes while supporting job-seeker–oriented functionality.

- C. Data Preparation

After retrieval, resume PDFs are processed using PyPDF2 to extract raw text. This unstructured content is prepared for parsing by dividing the text into logical sections such as contact information, education, experience, and skills. Lang Chain manages the text segmentation and metadata tagging required for consistent formatting before parsing.

- D. Text Preprocessing

To enhance accuracy and reduce errors in parsing, resumes undergo basic text preprocessing which includes:

1. Removal of special characters, headers, and footers.
2. Standardization of spacing, case formatting, and label tagging.
3. Filtering of irrelevant content such as resume templates or watermarks.

These steps ensure that the data passed to the LLM is clean and semantically consistent.

- E. Tokenization and Prompt Encoding

Lang Chain handles chunking of long documents and prepares tokenized prompts suitable for GPT-3.5 API requests. This includes creating templates for extracting specific fields (e.g., skills, experience) and chaining prompts to handle large resumes exceeding token limits. Output is captured in structured formats such as JSON and CSV.

- F. Resume Screening and Candidate Scoring

Each parsed resume is evaluated using a rule-based scoring model defined in an external JSON configuration file (scoring_config.json). The model includes weighted criteria such as:

1. Skill Matching – Direct and fuzzy match with predefined skills.
2. Project Relevance – Semantic similarity of candidate projects to role expectations.
3. Certifications and Education – Bonus scoring based on professional qualifications.
4. Effort Index – Penalty for training needs or onboarding complexity.

The results are exported to candidate_ranking_metrics.csv for ranking and reporting.

- G. Automation and Dashboard Integration

The system automates the flow from scoring to communication and analytics:

- Scored data is visualized in Power BI dashboards customized for HR, interviewers, and business leads.
- Automated emails are triggered through Gmail API, delivering application status or interview instructions.

- Logs track pipeline health and email delivery metrics.

- H. Post-Processing Outputs

After parsing and scoring, the system performs result refinement including:

1. Grammar and structure corrections in GPT summaries.
2. Merging of split sections or misclassified fields.
3. Final output validation and score normalization.

This ensures professional output for HR use and traceability in scoring logic.

- I. User Interface and Dashboards

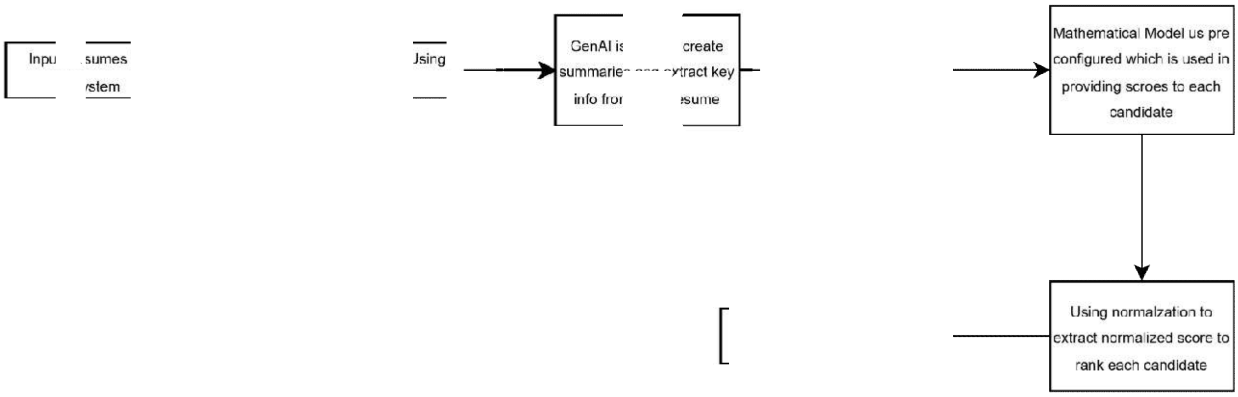
Instead of a custom-built UI, the system leverages Microsoft Power BI to offer user-friendly, interactive dashboards. These dashboards allow HR users to:

- Sort and filter candidate rankings.
- Review parsed resume data and score breakdowns.
- Export or share insights across departments.

This minimizes the development overhead while providing rich visual reporting.

- J. Agile Development and Continuous Integration

The system was developed following the Agile Software Development Lifecycle (SDLC). Iterative sprints were used to incrementally develop modules such as resume parsing, scoring logic, and dashboarding. Regular stakeholder reviews and daily stand-ups enabled quick feedback integration and reduced development friction. Continuous testing and logging ensured module-level validation throughout development.


Inpu sumes
/stem
Jsing
GenAI is create
summarie- --- -xtract key
info fror esume
Mathematical Model us pre configured which is used in providing scroes to each candidate
Using normalzation to extract normalized score to rank each candidate

IV. EXPERIMENTAL RESULTS

To evaluate the efficacy of the proposed Cognitive Document Intelligence Engine, a series of controlled experiments were conducted to simulate a real-world recruitment workflow. The objective was to assess the performance of the system across key modules—namely resume parsing, candidate scoring and ranking, dashboard integration, and communication automation.

The system was tested using a dataset of anonymized resumes submitted via email. These were processed end-to-end by the pipeline, which included parsing using GPT-3.5 via Lang Chain, scoring using a predefined configuration file, ranking based on cumulative scores, and visualization via Power BI. Email notifications were also dispatched to simulate real-time communication with candidates and internal stakeholders.

Table I provides a summary of the performance metrics collected during evaluation.

Table I: System Performance Evaluation

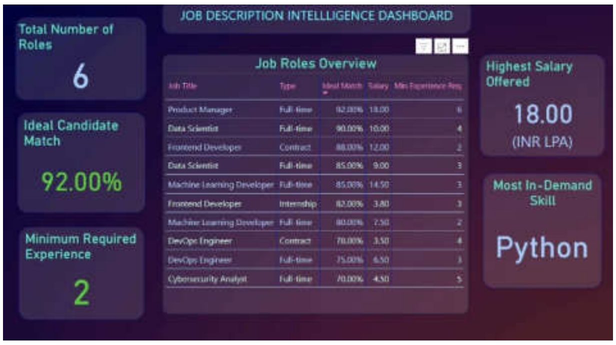
Metric	Observed Value	Description
Resume Parsing Accuracy	91%	Percentage of resumes with correctly extracted fields (skills, experience, etc.)
Candidate Ranking Agreement	85%	Agreement between system-generated and manually reviewed rankings
Dashboard Load Time	< 5 seconds	Average time for Power BI to refresh with ~50 candidate records
Email Dispatch Success Rate	> 95%	Percentage of emails sent successfully using the Gmail API
End-to-End Processing Time	~18 seconds/resume	Time from email receipt to dashboard-ready output including logging

The parsing accuracy metric was calculated by manually verifying a subset of resume outputs against ground-truth expectations. Candidate ranking alignment was evaluated through comparison with HR-generated shortlists, and the dashboard's responsiveness was measured during interaction-based filtering tests.

These results demonstrate the system's operational reliability and real-time responsiveness. The integration of GPT-3.5 with LangChain resulted in high accuracy for semantic parsing of resume content. The modular CSV-based scoring and ranking pipeline ensured transparency and traceability of results.

While formal NLP evaluation metrics such as ROUGE or BLEU were not applicable due to the non-generative nature of the scoring output, the system's performance was validated through functional correctness and feedback from trial users, including HR professionals and technical reviewers.

The framework's ability to automate resume intake, produce ranked evaluations, visualize candidate metrics, and send email updates with minimal human intervention illustrates its practical value in recruitment automation.



JOB DESCRIPTION INTELLLIGENCE DASHBOARD

Total Number of Roles: 6

Ideal Candidate Match: 92.00%

Minimum Required Experience: 2

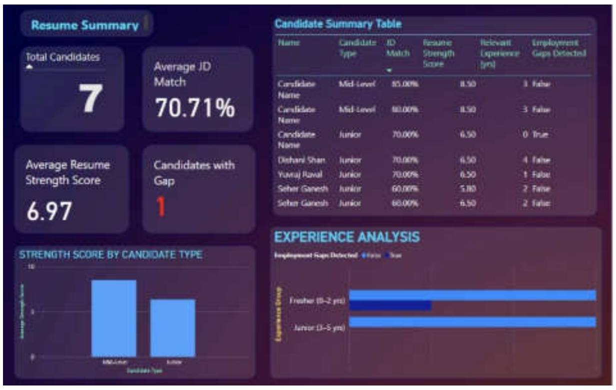
Highest Salary Offered: 18.00 (INR LPA)

Most In-Demand Skill: Python

Job Roles Overview

Job Title	Type	Ideal Match	Salary	Min Experience Req
Product Manager	Full-time	92.00%	18.00	6
Data Scientist	Full-time	90.00%	10.00	4
Frontend Developer	Contract	88.00%	12.00	2
Data Scientist	Full-time	85.00%	9.00	3
Machine Learning Developer	Full-time	85.00%	14.50	3
Frontend Developer	Internship	82.00%	3.80	3
Machine Learning Developer	Full-time	80.00%	7.50	2
DevOps Engineer	Contract	78.00%	3.50	4
DevOps Engineer	Full-time	75.00%	6.50	3
Cybersecurity Analyst	Full-time	70.00%	4.50	5

Power BI Dashboard(I)



Resume Summary

Total Candidates: 7

Average JD Match: 70.71%

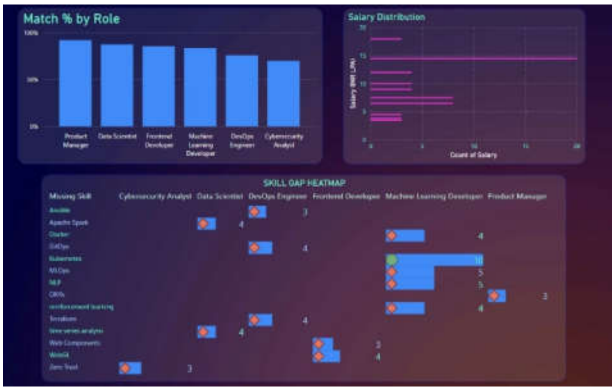
Average Resume Strength Score: 6.97

Candidates with Gap: 1

Candidate Summary Table

Name	Candidate Type	JD Match	Resume Strength Score	Relevant Experience (yrs)	Employment Gaps Detected
Candidate Name	Mid-Level	85.00%	8.50	3	False
Candidate Name	Mid-Level	80.00%	8.50	3	False
Candidate Name	Junior	70.00%	6.50	0	True
Dishani Shan	Junior	70.00%	6.50	4	False
Yuvraj Raval	Junior	70.00%	6.50	1	False
Seher Ganesh	Junior	60.00%	5.80	2	False
Seher Ganesh	Junior	60.00%	6.50	2	False

Power BI Dashboard(II)



SKILL GAP HEATMAP

Missing Skill	Cybersecurity Analyst	Data Scientist	DevOps Engineer	Frontend Developer	Machine Learning Developer	Product Manager
Ansible			3			
Apache Spark		4				
Docker					4	
GitOps			4			
Kubernetes					10	
MLOps					5	
NLP					5	
OKRs						3
reinforcement learning					4	
Terraform			4			
time series analysis		4				
Web Components				3		
WebGL				4		
Zero Trust	3					

Power BI Dashboard(III)

V. CONCLUSION

In this study, we presented a modular and AI-driven system designed to automate the resume screening and candidate ranking process. By leveraging advanced language models, including Open AI's GPT-3.5 via Lang Chain, the system efficiently extracts structured insights from unstructured resume data. This transformation significantly reduces manual workload and enhances consistency in candidate evaluation. The framework integrates seamlessly with automation tools such as the Gmail API and visualization platforms like Power BI, enabling real-time communication, transparent scoring, and interactive reporting. The backend logic is implemented through modular Python scripts and structured configuration files, supporting both maintainability and scalability.

While the current implementation relies on CSV-based data flow and REST ful API interactions, the architecture is extensible. Future integration with micro service frameworks such as Django or Spring Boot can further support distributed deployments and multi-user environments.

Experimental results validate the system's effectiveness across metrics such as parsing accuracy, ranking reliability, and communication success rate. These findings suggest that AI can play a transformative role in recruitment and resource optimization, offering substantial improvements in operational productivity.

In conclusion, the Cognitive Document Intelligence Engine demonstrates how the application of natural language processing and automation can streamline talent acquisition workflows, support informed decision-making, and lay the foundation for future advancements in intelligent HR systems.

REFERENCES

[1] Muresan, S., Tzoukermann, E. and Klavans, J.L., 2001. Combining linguistic and machine learning techniques for email summarization. In Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning (ConLL).

[2]Yousefi-Azar, M. and Hamey, L., 2017. Text summarization using unsupervised deep learning. Expert Systems with Applications, 68, pp.93-105.

[3] Biswas, P.K. and Iakubovich, A., 2022. Extractive summarization of call transcripts. IEEE Access, 10, pp.119826-119840.

[4] Suanmali, L., Salim, N. and Binwahlan, M.S., 2009. Fuzzy logic based method for improving text summarization. arXiv preprint arXiv:0906.4690.

[5] Manojkumar, V.K., Mathi, S. and Gao, X.Z., 2023. An Experimental Investigation on Unsupervised Text Summarization for Customer Reviews. Procedia Computer Science, 218, pp.1692-1701.

[6] Yang, X., Li, Y., Zhang, X., Chen, H. and Cheng, W., 2023. Exploring the limits of chatgpt for query or aspect-based text summarization. arXiv preprint arXiv:2302.08081.

[7] Xu, S., Zhang, X., Wu, Y. and Wei, F., 2022, June. Sequence level contrastive learning for text summarization. In Proceedings of the AAAI conference on artificial intelligence (Vol. 36, No. 10, pp. 11556-11565).

[8] Laskar, M.T.R., Hoque, E. and Huang, J.X., 2022. Domain adaptation with pre-trained transformers for query-focused abstractive text summarization. Computational Linguistics, 48(2), pp.279-320.

[9] J. Ochmann, S. Laumer, "AI Recruitment: Explaining job seekers' acceptance of automation in human resource management." In Wirtschaftsinformatik (Zentrale Tracks),2020,pp. 1633-1648.

[10] S. Strohmeier, F. Piazza, "Artificial intelligence techniques in human resource management—a conceptual exploration," Intelligent Techniques in Engineering Management: Theory and Applications,

2015, pp. 149-172.
[11] K. Wailthare, A. Tamhane, V. Mulik, K. Suryawanshi. "A Cosine Similarity based resume screening system for job recruitment,"
International Research Journal of Modernization in Engineering
Technology and Science, 5(4), (2023): 1840 - 1844
[12] K. Zechner, ''A literature survey on information extraction and text
summarization,'' Comput. Linguistics Program, Carnegie Mellon Univ.,
Pittsburgh, PA, USA, Tech. Rep., Apr. 1997. [Online]. Available:
http://www.cs.cmu.edu/~zechner/infoextr.pdf
[13] H. P. Luhn, ''The automatic creation of literature abstracts,'' IBM J. Res.
Develop., vol. 2, no. 2, pp. 159–165, Apr. 1958.
[14] J. Kupiec, J. Pedersen, and F. Chen, ''A trainable document summarizer,''
in Proc. 18th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., 1995,
pp. 68–73.
[15] J. M. Conroy and D. P. O'leary, ''Text summarization via hidden Markov
models,'' in Proc. 24th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf.
Retr., 2001, pp. 406–407.
[16] E. Mittendorf and P. Schauble, ''Document and passage retrieval based on
hidden Markov models,'' in Proc. 17th Annu. Int. ACM SIGIR Conf. Res.
Develop. Inf. Retr., 1994, pp. 318–327.
[17] F. Chen, K. Han, and G. Chen, ''An approach to sentence-selectionbased text summarization,'' in Proc. IEEE TENCON Conf., Oct. 2002,
pp. 489–493.
[18] Y. Gong and X. Liu, ''Generic text summarization using relevance measure
and latent semantic analysis,'' in Proc. 24th Annu. Int. ACM SIGIR Conf.
Res. Develop. Inf. Retr., 2001, pp. 19–25.
[19] Z. Wu, R. Koncel-Kedziorski, M. Ostendorf, and H. Hajishirzi,
''Extracting summary knowledge graphs from long documents,'' 2020,
arXiv:2009.09162.
[20] N. Franciscus, X. Ren, and B. Stantic, ''Dependency graph for short
text extraction and summarization,'' J. Inf. Telecommun., vol. 3, no. 4,
pp. 413–429, Apr. 2019.
[21] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang, ''Extractive
summarization as text matching,'' in Proc. 58th Annu. Meeting Assoc.
Comput. Linguistics, 2020, pp. 6197–6208.
[22] J. Neto, A. A. Freitas, and C. A. Kaestner, ''Automatic text summarization
using a machine learning approach,'' in Proc. Brazilian Symp. Artif. Intell.
Berlin, Germany: Springer, 2002, pp. 205–215.
[23] K. Kaikhah, ''Automatic text summarization with neural networks,'' in
Proc. 2nd Int. IEEE Conf. Intell. Syst., Jun. 2004, pp. 40–44