ADROID Conference Series: Engineering and Technology
Volume 2, Issue 3, pp.118-124, 2026

# An AI-Powered Resume Analysis Suite

Waseem Ahmed, Manish Rajora, Hemant Kumar, Naman Negi

Department of CSE-AIML, ABES Engineering College, Ghaziabad, India

waseemahmed0609@gmail.com, manish.22b1531078@abes.ac.in, hemant.22b1531067@abes.ac.in, naman.22b1531011@abes.ac.in

**Abstract.** DeepMatch is a resume analysis application built with AI that enables Applicant Tracking Systems (ATS) to converse with job applicants. It has an accessible, open resume comparison forum that presents resumes side by side, as an appealing job study. particular job descriptions. Fire at your resume through higher-level NLP and AI systems, including NLTK, Spacy and Lang chain, the LangGraph - DeepMatch retrieves, formats, and dissects the text of your resume so that it can locate the key requirements that are absent, calculate your own ATS compatibility measurements and offer suggestions for personal aspects of skill building. DeepMatch will give you personalised assistance on which route to upgrade in your occupation, how to improve your oeuvre before publishing it, and on discovering ability. When deployed on Streamlit Cloud, ensure it is available and functions efficiently. DeepMatch assists the candidate more closely towards their desires as the AI input to the idea of the candidate is combined with the precision of NLP, meeting the company requirements to better quality results, where the resumes can be more easily comprehended, and the procedures of automated recruitment are clearer.

***Keywords:*** Artificial Intelligence, Resume Analysis, ATS, NLP, LangChain, LangGraph, Streamlit, AWS Bedrock, Skill Gap, Generative AI and Analysis

## 1. INTRODUCTION

One hundred and thousands even millions of applications are offered to each vacancy there is in the job market today. To counter this rush of applications, companies have adopted ATS to filter the applicants. It represents a narrow AI which is more focused on assessment of job requests and resumes. However, these systems may as well also exhibit the characteristics of a black box to reject a good candidate due to simple formatting errors, an unconventional current job title or box, passing less qualified candidates above the well qualified ones due to differences in much finer grained factors such as: name spelling errors, which name is used, etc. It is an enormous issue among individuals seeking employment, and even unprecedented to realize it. why their applications were rejected without a single individual examining their resume. The applicants whose skills match exactly to a different word or phrase will be penalized because most of the archaic systems of tracking applicants pay attention to the perfect results in terms of the exact word search. At the same time NLP and AI have evolved to create powerful means that we can employ to comprehend and even perceive the languages of people.

These technologies act as the solution to making the process of tracking applicants demystified, i.e., by creating the proximity between the candidates These technologies are generated to facilitate demystification of the applicant tracking system in the shape of qualification and machine-readable resumes. An exclusive set to offer tailored and useful feedback continues miles behind even in spite of the number of applications to create resumes and identify keywords. There are lots of technologies available today that claim to cut keywords- that cuts no ice in the way of giving the semantic context and positive or negative advice that a candidate would actually require to be improved. They can know what is missing, and they cannot answer why, they cannot give a solution. To solve this problem, we have offered DeepMatch, an AI based resume analysing engine. DeepMatch tries to provide job seekers with their empowerment, which consists in tricking the review of the application tracking system, and comparing the resume of a candidate against the description provided of a particular job. The main contribution of our system is the combination of two AI paradigms:
1) Traditional NLP (NLTK, spaCy) for accurate, fast extraction and scoring, and
2) Generative AI (via LangGraph and AWS Bedrock) for contextual, human-like guidance.
The system employs generative AI to provide targeted recommendations for skill development in addition to determining an ATS compatibility score and finding critical missing keywords. This paper describes DeepMatch's system architecture, its AI and NLP pipeline approach, the outcomes of its analysis capabilities, conclusions, and future developments.

## 2. SYSTEM ARCHITECTURE AND TECHNOLOGIES

The architecture of DeepMatch is intended to be a contemporary distributed web application. It is a modern web application since it isolates the user interface from the complex AI backend. The user interface is kept quick and responsive thanks to this decoupled design, and a scalable, serverless backend manages the computationally costly AI processing. Additionally, this paradigm permits the independent updating of individual components. Several significant technologies are included into the system.

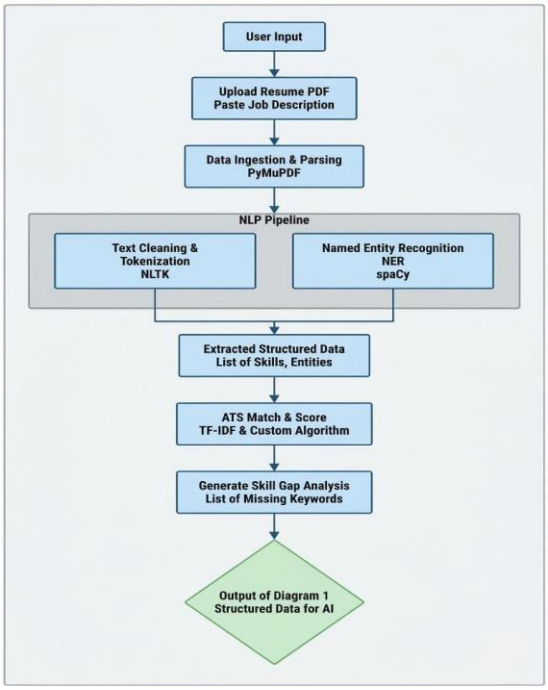
*A. FrontEnd (Streamlit)*

The frontend is developed using Streamlit, an open-source Python framework. This decision was made on purpose since it allowed for the rapid creation of an interactive, data-driven web interface without the need for knowledge of conventional frontend languages (such JavaScript or React). This is perfect for projects with Python-based core logic and for quick prototyping.

We use a number of essential Streamlit components in the application. Users can upload their PDF resumes using the st.file_uploader widget's straightforward interface. The free-form job description text is captured using the st.text_area.The user may delve down into their score, the keyword analysis, and the AI recommendations because all results are displayed using st.expander containers.We leverage Streamlit's st.session_state functionality, which saves data while the user interacts with the application, to handle user reports over numerous runs.

*B. BackEnd(Python, LangChain and LangGraph)*

The complete backend logic is powered by Python, which is used as a common language for web serving, NLP and AI coordination.




**LangChain:** We have used LangChain's architecture to build AI powered tools particularly to build tools which respond to prompts and communicate with language models. We can create sustainable chains as LangChain acts as coordination layer for this tool.

Example, a basic chain include: (i) skill extraction prompt; (ii) model to carry the prompt; (iii) output parser to convert model's response into python list.

**LangGraph:** A straightforward linear chain is inadequate for the multi-step analysis procedure. We make use of LangGraph, a framework for creating multi-actor, stateful AI systems. This makes it possible to create complex, stateful AI systems that function as a graph. This is crucial for the movement of data between stages (e.g., from text extraction to scoring to suggestion). The workflow of our agent is a cycle rather than a single line: it parses text, scores it, and then conditionally chooses which tool to use next (e.g., "generate recommendations" or "ask for clarification"). As it traverses this graph, LangGraph controls the state (the resume content, skill gap, etc.).

*C. NLP Pipeline (NLTK and spaCy)*

An crucial component is the NLP pipeline, which manages the unstructured text from resumes and job

descriptions. We are able to base the AI with structured data because this pipeline operates prior to any creative AI calls. We employ:

**NLTK (Natural Language Toolkit):** Basic pre-processing activities are performed with this library. Text that is taken out of a PDF is frequently "dirty". To divide the text into digestible chunks, utilise NLTK's word_tokenize and sent_tokenize functions. Next, we use NLTK's corpus.stopwords to apply a stop-word removal technique to exclude common words that create noise to keyword analysis, such as "and", "the", and "is".

**spaCy:** We leverage spaCy's high-performance, pre-trained statistical models for more complex NLP characteristics. Named Entity Recognition (NER) is its main use in DeepMatch. To effectively parse the resume and tag entities like SKILL (Python, Java, Matplotlib), ORG (company and university names), and DATE. Compared to keyword matching alone, this structured data offers a far more accurate inventory of the candidate's skills.

*D. Data Visualization (Matplotlib)*

DeepMatch use Matplotlib library to generate visualisations that make analysis results clear, easy to understand and effective. These charts highlights skill gaps, display ATS score and provide a clear picture of resume's pros and cons. On the moment, this tool create horizontal bar chart that shows matched keywords with missing keywords providing a brief overview of candidate's/user's fit for position.

*E. Cloud and Deployement (AWS Bedrock and Streamlit Cloud)*

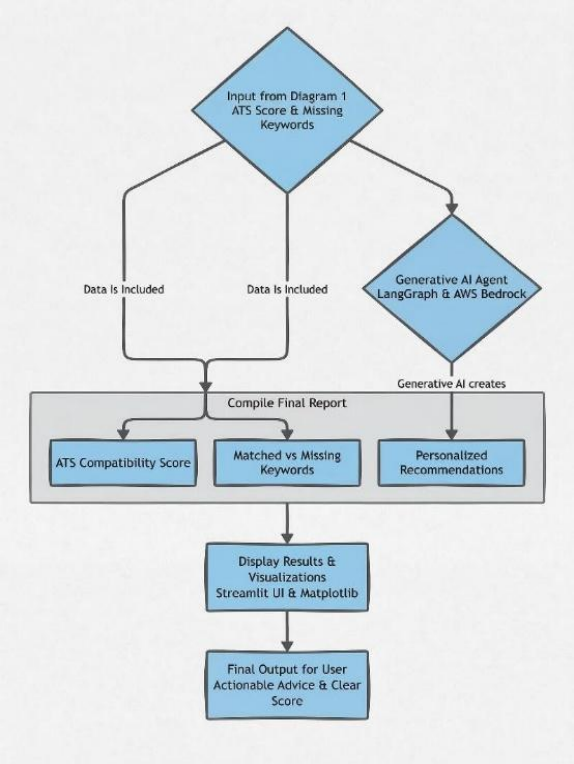
We have made a crucial architecture choice by separating AI model from application hosting.

Streamlit Cloud: The tool is hosted on Streamlit Cloud for easy access. Streamlit's direct Github integration made continuous deployement possible. Whenever we have a push to the main branch the application instantly updated with those changes, which is perfect of academic project development.

AWS Bedrock: The main AI model is hosted on AWS Bedrock which provide a scalable and secure endpoint. This serverless strategy offers a number of benefits: (1) It eliminates the requirement to manage or provide servers. (2) Because the big model (billions of parameters) is not included with the app, it makes our application lightweight. (3) It makes it simple to try out and switch between various foundation models to determine which one works best for our particular goal.

## 3. METHODOLOGY

The DeepMatch suite is used to guide the user through a resume optimising workflow. The entire procedure is intended to be a stateful, iterative loop, from data intake to recommendation.




*A. Resume Analysis and Preview*

The backend initiates a parsing process whenever the resume file in the form of a pdf is uploaded on the system. To ensure that the raw text blocks get extracted in the original document structure and to a high percentage of accuracy we have relied on the PyMuPDF module. This crude text is passed at once to our NLP pipe (discussed in Section II-C). NLTK and spaCy clean and structure the text that also extract the significant parts and entities. Through display: ""feature Resume Preview" of the text extracted, clean as well as the original PDF (built within an iframe), the user interface give the user the ability to check whether the data was properly processed.

*B. ATS Match & Score*

This is the primary feature of DeepMatch. The NLP pipeline immediately evaluates a job description that the user uploads in order to extract critical skills, technologies, and credentials.

This list of required keywords is then compared to the user's parsed resume content.

Custom-weighted keyword matching method used in the scoring algorithm:

1. JD Points Extraction: A TF-IDF analysis is then used to locate the most significant n-grams (one or two word phrases) after tokenising the text of the job description, eliminating typical stopwords. This ends up with a vector of "required skills".
2. Resume Skill Extraction: Our spaCy NER model processes the resume text to extract all detected SKILL entities, resulting in a vector of "candidate skills".
3. Scoring: A compatibility score is determined by calculating the percentage of matched terms, weighted by their relevance (e.g., "required" vs. "preferred" if identified). This is determined as a percentage: (Total Required Keywords/Number of Matched Keywords) * 100.
4. Gap Analysis: The system calculates the specified difference between the candidate's talents and the necessary skills. The user is subsequently given a clear list of the missing terms, which serves as the input for the following step.

*C. Skill Recommendations*

The first step is to identify a skill gap. To provide useful advice, the list of missing keywords is put into an advanced AI workflow made with LangGraph. This procedure queries a generative AI model using AWS Bedrock.

DeepMatch's primary innovation is this functionality. A graph-based agent with several nodes is defined as

follows:

- Parse_inputs: This node receives the job description and resume in their raw form.
- Run_nlp_pipeline: This node extracts job requirements and structured skills by running the NLTK/spaCy analysis.
- Calculate_score_and_gap: This node uses Section III-B's scoring approach. It adds the score and the missing_skills list to the graph's state object.
- produce recommendations: The generating node is this one. The complete resume content and the missing_skills list are used as context. It applies a sophisticated prompt to the AWS Bedrock model, like: "You are a useful career mentor." The user lacks the following job-related abilities: [skills_missing].

This is their resume: [resume_text]. Please offer three to five practical suggestions. Check to see if they have a relevant skill (Tableau is related to Power BI, for example) for each missing skill, and offer suggestions for rephrasing their expertise. If they don't have any relevant skills, recommend a modest project or certification they may obtain. The phrase Before making customised recommendations, the model is told to look at the user's past experience and the abilities they lack. These are more than just broad descriptions; they can provide ideas for rephrasing existing knowledge or creating new skills.

*D. Session History and Visualizations*

The outcomes of the analysis are saved in the form of a report; it has included the score, missing keywords, recommendations, and Matplotlib charts. They are also characterized by the aspect of Session History in which a user can view his history of reports, or mass download his history of reports in a bid to analyze his progress occasionally. This is achieved by means of session state which is a short-term storage of memory that remain active until the web browser tab of the user is active.

## 4. RESULTS AND DISCUSSION

In order to check the algorithm, we have contrasted it with a number of other resumes of real job descriptions. As a result of the comparison of ourpdf resumes with various types of format the system gave several satisfactory results.

*A. Case Study: B.Tech Student vs Data Analyst Role*

To do a case study we checked the accuracy of the system using real JD of data analyst role with sample resume.

- Input:
  - Resume analysis: The resume contains excellent points related to Python, SQL, Pandas and Tableau showing the works in Data Cleaning and Visualisation.
  - Job Description points: The JD demands experience in Python, SQL, Power BI and communication.
- Results:
  - ATS Score: DeepMatch presents results that are comparable to 65% ATS score,
  - Keyword Analysis: Python and SQL were entered into system as keywords.
  - Visualizations: The matched and deficient skills are easily seen through horizontal bar charts.

*B. Discussion of Benefits and Use Cases*

This case study presents the main advantages of Deepmatch that is represented as-

- Better Quality of the Resumes: The Resumes that were presented as a concise and to the point report to the user was better in quality because missing keywords were the key words which rated 60 points.
- Developed, Specific Action Recommendations: The AI model, in fact, actually offer professional guidance as opposed to matching keywords. It is the contextual guidance that is the most valuable input of our approach.
- Greater Efficiency: The automated process helps the job-seekers to customize their resumes to suit various applications and maximize their output and probability of success.

The primary purpose of the application is to help job seekers enhance their resumes. Recruiters and career counsellors can utilise the app to speed up candidate screening and provide structured feedback, expanding their capacity to assist more students. This is a secondary use case.

### *C. Limitations*

During testing and development, we found certain limitations. The accuracy of the NLP pipeline depends on conventional resume forms; highly styled or graphical resumes created using programs like Canva may make text extraction challenging or disorganised. Furthermore, AI's recommendations are only suggestions, and the user's ultimate decision determines whether or not they are successfully implemented. Currently, the system only supports PDF file uploads; it does not handle the popular format.docx files. Lastly, a persistent, user-account-based system would be a big improvement because the session history is erratic and disappears as the browser tab is closed.

## 5. CONCLUSION AND FUTURE DEVELOPMENT

DeepMatch is now a state-of-the-art architecture powered by AI and enabling resumes and job seekers to genuinely be empowered. It provides skill suggestions and skill to act on and ATS compatibility score, and individual resume evaluation. The technology provides the candidates with an enhanced appearance on how to enhance their resumes and clear the ATS screening process. The major change of its contribution is its two process levels an accurate NLP score generator and active generative AI advisory in case of contextual, human-like advice.

For future development, we want to focus on two primary areas.

1. Integration and Data Expansion: First was integration with employement boards. Rather than copy job description, we use the API’s from job websites like LinkedIn. In order to accommodate a wider user base, we also want to look into multilingual resume analysis and expand support for multiple file formats utilize different modules in python.
2. Enhanced AI Capabilities: we will concentrate over ongoing model improvement. We intend to switch from Bedrock's general-purpose models to a refined model that was trained on a particular dataset of successful resume-to-job-description pairings. A "Resume Re-writer" module will be a major addition in the future. To further bridge the gap between analysis and action, this agent would not only recommend modifications but also create new resume bullet points based on its suggestions, which the user could then evaluate and approve.

**Conflict of Interest**

The authors declare no conflicts of interest regarding the current research.

**References**

[1] Systematic Review of Resume Analysis Methods, (at at least) D. Kanikar, P. Jain, S.Jain, and L.Purohit, in Proc. 2025 International Conference on Computer Science and Communication Engineering (ICCSCE), 2025. doi:10.2991/946463858-275.

[2] A. Heakl, Y. Mohamed, N. Mohamed, A. Elsharkawy, and A. Zaky, ResumeAtlas: Revisit Resume Classification with Large-scale Both Datasets and Large Language Models, in Neural and Language Processing, Proc. 6 th Int'l Conf. on AI in Computational Linguistics, 2024. DOI:10.48550/ARXIV.2406.18125.

[3] S. Deepak, S. R. Rubeesh, D. Kabilan and V. Sathya, Automated resume screening using the might of ML and NLP to screen and rank applicants, in Proc. Int'l Conf. on Emerging Trends in AI and Computation (ICONEST), 2026. doi:10.1063/5.0308593.

[4] Y. Huang, D.-R. Liu and S.-J. Lee, Talent recommendation with the assistance of attentive deep neural network and implicit resume relationship, Information Processing and Management, vol. 60, no. 4, 2023, doi:10.1016/j.ipm.2023. 103357.

[5] N. Jahan, M. B. Uddin, M. S. Arefin, C. M. Mehedi, F. Tasnim and K. E. Delowar, The use of Multi-stage LLM Classification and Hybrid With Within the process of Resume Screening, in Proc. 2025

[6] R. V. K. Bevara et al., Resume2Vec: Intelligent Resume Embeddings to the applicant tracking system toeven revenue Prominent Mathews, Electronics, vol. 14, no. 4, 2025. doi:10.3390/electronics14040794.

[7] NLP and AI-based efficient screening of candidates A. Upadhye, Automating Resume Classification, International Journal of Computer Applications, vol. 185, no. 40, pp. 46-50, Nov. 2023.

[8] A. Hepzibah R. et al., AI-Powered resumes Screening System: contingent hiring NLP and Large Language models: effective and fair Resumes, J. Theoretical and Applied Information Tech., vol. 103, no. 23, pp.

10146-10155, Dec. 2025.
[9] P. Frazzetto, M. U. Haq, F. Fabris and A. Sperduti, Graph Neural Networks in the Candidate-Job Matching: an Inductive Learning Algorithm, Data Science and engineering, 2025, doi: 10.1007/s41019-025-00293-y.
[10] R. S. Pundir et al., "Improvement of Resume Recommendation System via Skill based similarity Recommendation with the help of deep learning Models, In Proc. 7th Inventive Computation Technologies (ICICT) 2024.doi:10.1109/ICICT60155.2024.10544875.
[11] Zhu et al., How to: Job Recommendations with the LLM-based Resume Completion: Information Processing and Management, no. 62, no. 6, 2025, Article 104261.doi:10.1016/j.ipm.2025.104261.
[12] S. Luo, J. Yu, ESGNet: A multimodal network model, on the entity semantic graphs, extract the information on Chinese resumes, Information Processing and Management, vol. 61, no. 1, 2024, Article 103524. doi:notital there 103524.
[13] The article presents the research carried out by F. T. Tang et. al, Explainable person-job recommendation: challenges, approaches and comparative analysis, Frontiers in Artificial Intelligence, vol. 8, 2025, Article 1660548. doi:10.3389/frai.2025.1660548.
[14] A. C. T. Tuan et al., O ontology and its application in skills matching when recruiting jobs, Applied ontology, vol. 19, no. 1, 2024, pp. 1-20. doi:10.3233/AO-240019.
[15] A. Qostal, A. Moumen and Y. Lakhrissi, Resumes Classification using Neural Network with BERT and Gensim: CVs of Moroccan Engineering students, Article 74, Data, vol. 9, no. 6, 2024, p. 103390/data9060074.