Scalable resume screening using large language model Meta AI version 3

Asmita Deshmukh[1,2], **Anjali Raut**[1], **Vedant Deshmukh**[3]
[1]Department of Computer Science and Engineering, HVPM College of Engineering and Technology, Amravati, India
[2]Department of Computer Engineering, K. J. Somaiya Institute of Technology, Mumbai, India
[3]Department of Computer Engineering, Sardar Patel Institute of Technology, Mumbai, India

Article Info

Article history:

Received Jul 5, 2025
Revised Dec 30, 2025
Accepted Jan 22, 2026

Keywords:

Automated recruitment
Large language models
LLaMA 3
Natural language processing
Resume screening

ABSTRACT

This research paper explores the use of large language model Meta AI 3 (LLaMA 3) for automating the resume screening process. Traditional resume screening methods that rely on keyword searching and human review can be inefficient, biased, and fail to identify qualified candidates. LLaMA 3, trained on large-scale text datasets, has the potential to accurately analyze resumes by understanding context and semantic details beyond simple keyword matching.The study presents a system that converts resume PDFs to text, inputs the text along with the job description into the LLaMA 3 model, and generates a ranked list of candidates with reasoning for their job fit. This discusses the data preparation, model setup, and performance evaluation of this system. Results show LLaMA 3 can rapidly process batches of resumes while reducing human bias in the screening process. The system aims to streamline hiring by automating the initial resume screening stage to surface top candidates for further in-depth evaluation. Key benefits include improved accuracy in identifying relevant skills, reduced bias compared to human screeners, and significant time savings for recruiters. The paper also examines ethical considerations around using AI for hiring decisions. Overall, this work demonstrates the promising application of large language models (LLMs) like LLaMA 3 to transform and enhance resume screening practices.

Corresponding Author:

Asmita Deshmukh
Department of Computer Science and Engineering, HVPM College of Engineering and Technology
Amravati, Maharashtra, India
Email: asmitadeshmukh7@gmail.com

1. INTRODUCTION

Automated resume screening has emerged as a critical area of interest in recruitment, aiming to address the inefficiencies and biases inherent in traditional manual review methods. A pioneering system for extracting information from unstructured resumes using natural language processing (NLP) techniques marked an initial step toward streamlining the screening process [1]. The effective use of rule-based NLP to extract housing status from clinical narratives highlighted the potential of utilizing contextual cues in unstructured text for identifying social determinants of health [2]. The use of NLP for predictive modeling of HIV diagnoses highlights the capability of these techniques to extract meaningful information from complex textual data [3]. The versatility of NLP across domains shows its use for accurately calculating adenoma detection rates in screening colonoscopies and for large-scale labeling of clinical records [4], [5]. These approaches reliance on domain-specific, rule-based, or traditional techniques limits the ability to capture nuanced context and generalize to diverse, unstructured data like resumes. The integration of machine learning algorithms such as support vector machine (SVM) and random forest (RF) has enhanced resume screening by improving the accuracy and scalability of text classification. A two-level classification system was applied for extracting features from resumes, achieving 85% accuracy in heading prediction using the extreme gradient boosting (XGBoost) algorithm [6]. A cosine similarity-based system, as presented in [7], demonstrated high accuracy and efficiency in extracting relevant skills and qualifications. An SVM-based model improved hiring efficiency by matching resumes with job descriptions [8], while a three-phase NLP-driven approach enhanced semantic resume analysis and candidate ranking [9]. The use of k-nearest neighbors (KNN) and SVM was explored for classifying candidates and resumes [10]. KNN classified candidates into different packages based on their profiles, while SVM categorized resumes into testing and development profiles. Machine learning techniques for resume screening are limited by the need for manual feature extraction, poor handling of unstructured resumes, and difficulty in processing long-form text and context.

A deep residual convolution long short-term memory (DR-CLSTM) network combines convolutional neural network (CNN) and long short-term memory (LSTM) to predict option prices more accurately and efficiently than traditional models like Black-Scholes by capturing complex market dynamics [11]. The study [12] explores NLP and LSTM-based models, showcasing improvements in resume classification accuracy across various applications. CNN and LSTM were used to diagnose arrhythmias from Electrocardiogram (ECG) signals [13], while combining Fourier-Bessel expansion with LSTM that achieved 90.07% classification accuracy [14]. A multi-layer recurrent neural network (RNN) improves skill and experience identification while integrating engagement surveys and business intelligence to support human resources (HR) decisions and team allocation [15].

Deep learning models for resume screening has limitations such as the need for pre-processing and embeddings, fixed context windows that hinder long resume handling, and difficulty capturing complex cross-paragraph relationships. Transformers, particularly bidirectional encoder representations from transformers (BERT), have revolutionized resume screening with their ability to capture bidirectional context. The study [16] uses BERT and named entity recognition (NER) to automate resume screening, improving accuracy and reducing manual effort. A BERT-based framework fine tuned on historical job application data is used to objectively predict person-job fit and improve resume screening efficiency [17]. Sentence-BERT (S-BERT) is widely used in automated resume screening for generating embeddings and improving relevance ranking via cosine similarity, often outperforming traditional BERT models in accuracy and efficiency through hybrid NLP approaches [18]–[20]. Sentence-pair BERT (SPBERT), a fine-tuned BERT variant, encodes resumes and job postings into unified embeddings to predict compatibility through adapted next-sentence prediction [21]. Transformer model is a powerful technique for resume screening, but are limited by challenges in zero or few shot classification, handling long texts, and requiring costly fine-tuning.

Large language models (LLMs) like generative pre-trained transformers (GPT), claude, and large language model Meta AI 2 (LLaMA 2) are increasingly used in abstract screening to automate evaluation through advanced language understanding and contextual analysis [22]. The paper [23] proposes an LLM-based resume screeninagent framework for resume screening that is 11 times faster than manual methods and achieves an 87.73% F1-score, surpassing GPT-3.5 in summarization, grading, and decision-making. LLM framework with retrieval-augmented generation (RAG) automates resume screening through context-aware extraction, evaluation, summarization, and scoring, closely aligning with HR assessments and enhancing recruitment scalability [24], [25].

Current resume screening method including rule-based systems, classical machine learning models, and early deep learning approaches are limited in handling unstructured and lengthy resumes due to their lack of contextual understanding, dependence on manual feature extraction, restricted context windows, and high fine-tuning costs. To overcome these constraints, there is a growing need for a scalable, adaptable, and context-aware solution capable of accurately evaluating resumes with minimal manual input. This study seeks to enhance resume screening by utilizing LLaMA 3 model’s capabilities to replace biased and inefficient keyword-based methods, enabling accurate candidate ranking, reduced human bias, and automation of the initial stages of hiring.

The remainder of this paper is organized as follows: section 2 details the methodology, including the data preparation process, PDF-to-text conversion, system configuration using LM Studio, and the design of the L3-based resume screening framework. Section 3 presents the results followed by section 4 discussion, covering model configuration, processing efficiency, comparative analysis of candidate evaluations, and performance benchmarking against other LLMs. Section 5 concludes the study, summarizing key findings, societal implications, and future scope with recommendations for ethical AI deployment in recruitment.

2. METHOD

This section outlines the methodological framework adopted to develop an automated resume screening system using the LLaMA 3 language model. It includes the formal problem formulation, data preparation, system architecture, model integration, and configuration settings. The approach emphasizes the use of contextual embeddings and prompt-based reasoning for accurate candidate-job matching, while ensuring fairness, scalability, and explainability.

2.1. Problem definition

Automated resume screening can be formally modeled as a multi-criterion ranking problem, where the objective is to match a set of candidate resumes to a job description based on predefined evaluation criteria while minimizing false positives and false negatives.

Let R, j, C be $R = \{r_1, r_2, \ldots, r_n\}$ be the set of candidate resumes; j be a job description with requirements $Q = \{q_1, q_2, \ldots, q_m\}$; $C = \{c_1, c_2, \ldots, c_k\}$ represent the set of evaluation criteria such as skills, experience, and qualifications. The goal is to learn a ranking function $f: R \times Q \rightarrow R$ such that for each resume r_i, the function $f(r_i, j)$ computes a relevance score based on textual similarity, contextual alignment, and domain-specific requirements. The optimization objective is to maximize candidate-job relevance while mitigating misclassification errors.

2.2. Data preparation and system architecture

This section describes the data preparation and system architecture implemented for the proposed automated resume screening system using the LLaMA 3 model. The process begins with converting unstructured resume documents, typically in PDF format, into structured plain text to enable downstream processing such as embedding generation, classification, and candidate ranking. To perform this conversion, LM Studio which is a specialized software platform for running LLMs. LM Studio supports local deployment of the LLaMA 3 model, enabling efficient and private execution without reliance on cloud-based APIs. The platform streamlines the transformation of resumes into analyzable text by facilitating precise segmentation of key sections such as education, experience, and skills.

2.2.1. Resume preprocessing

Resumes are typically received in PDF format, which is unstructured and unsuitable for direct machine processing. A preprocessing step is required to convert these resumes into machine-readable text. This was accomplished using a Python script developed with the PyPDF2 library. The script defines a function that accepts the file path of the input PDF and the desired output text file. It opens the PDF in binary mode, uses the PdfReader class to parse its contents, and extracts text page by page using the .extract_text() method. The resulting text is then written into a .txt file that preserves the original logical flow of resume components. These structured text files serve as the primary input to the LLaMA-based processing pipeline, ensuring that relevant information such as qualifications, skills, and work experience can be effectively interpreted by the language model.

2.2.2. System architecture overview

Figure 1 illustrates the system architecture developed for automated resume screening using the LLaMA 3 model. Once resumes have been converted to structured text and paired with the corresponding job description, both inputs are simultaneously processed through the LLaMA 3 model in a multi-stage pipeline. Initially, the inputs are transformed into contextual embeddings that capture semantic relationships between candidate profiles and job requirements. These embeddings are then subjected to root mean square (RMS) normalization to enhance stability during inference. Next, the model applies grouped multi-query attention with key-value caching, an optimization technique that allows the model to efficiently focus on the most relevant parts of the input across multiple resumes. Following attention, the output is passed through a feedforward neural network utilizing a switch gated linear unit (SwiGLU) activation function, which helps in modeling complex hierarchical relationships. This output is combined with the original embeddings and normalized again. Finally, a linear transformation is applied, and the resulting vectors are passed through a softmax layer to compute relevance scores. These scores represent the degree of alignment between each resume and the job description, enabling the system to rank candidates accordingly. This architecture enables high-resolution semantic matching and forms the backbone of the LLaMA-based ranking function.

2.2.3. Model deployment via LM Studio

The LLaMA 3 model employed in this study is the Meta-LLaMA-3-8B-Instruct variant, chosen for its optimal balance between model complexity and computational efficiency. Deployment was carried out using LM Studio, an open-source platform that supports local execution of LLMs without requiring cloud-based infrastructure. The model was configured with a quantization level of Q4_K_M, a context length of 8,192, and an embedding size of 4,096, enabling it to process long and information-rich input sequences effectively. Architecturally, it includes 32 transformer layers and 32 attention heads, with 8 heads designated for key-value caching, enhancing attention performance during inference. Positional encoding was handled using rotary position embedding (RoPE) with a frequency base of 500,000, which improves the model's ability to understand token order in long sequences.

To fine-tune the generation quality and diversity, temperature was set to 0.8, top-k sampling to 40, and a repeat penalty of 1.1 was applied to discourage repetitive outputs. For improved efficiency, GPU acceleration using NVIDIA CUDA was enabled, and CPU multithreading was optimized to reduce latency. The full model was loaded into RAM to support fast inference, and parameters such as prompt evaluation batch size and context window were adjusted based on system capabilities and data size. This configuration ensured smooth and efficient execution of the resume screening workflow while maintaining high accuracy in candidate-job matching. Figure 2 illustrates the complete configuration setup used in this study. Specifically, Figure 2(a) presents the model selection and initialization parameters, including the selected LLaMA 3 variant and prompt formatting options tailored for resume screening tasks. Figure 2(b) shows the hardware resource allocation, highlighting GPU core utilization and memory offloading settings adopted to ensure efficient inference. Figure 2(c) depicts the inference and hardware parameters, such as temperature, token generation limits, and sampling strategies, which collectively influence the model's response quality and consistency.

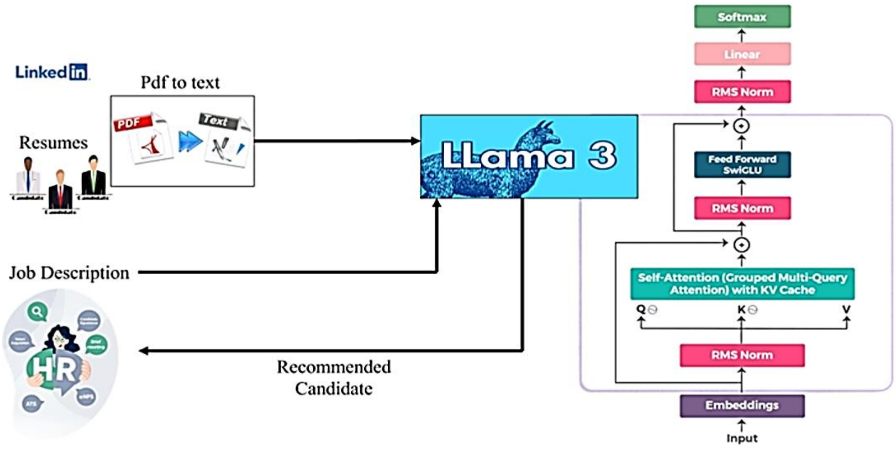
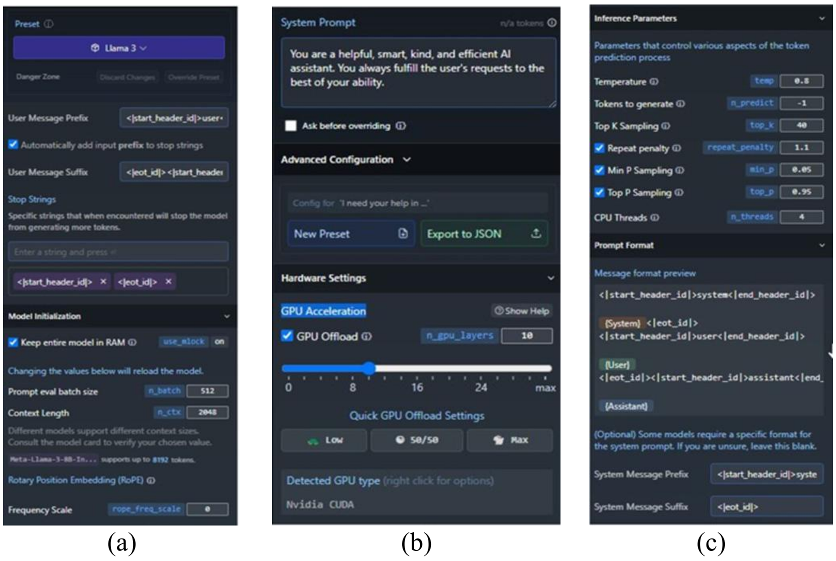


Figure 1. Proposed system architecture for automated resume screening using LLaMA 3 model


Preset
Llama 3
Danger Zone — Discard Changes — Override Preset
User Message Prefix: <|start_header_id|>user
☑ Automatically add input prefix to stop strings
User Message Suffix: <|eot_id|><|start_header
Stop Strings
Specific strings that when encountered will stop the model from generating more tokens.
Enter a string and press ⏎
<|start_header_id|> × <|eot_id|> ×
Model Initialization
☑ Keep entire model in RAM — use_mlock: on
Changing the values below will reload the model.
Prompt eval batch size — n_batch: 512
Context Length — n_ctx: 2048
Different models support different context sizes. Consult the model card to verify your chosen value.
Meta-Llama-3-8B-In... supports up to 8192 tokens.
Rotary Position Embedding (RoPE)
Frequency Scale — rope_freq_scale: 0

(a)

System Prompt n/a tokens
You are a helpful, smart, kind, and efficient AI assistant. You always fulfill the user's requests to the best of your ability.
☐ Ask before overriding
Advanced Configuration
Config for 'I need your help in ...'
New Preset — Export to JSON
Hardware Settings
GPU Acceleration — Show Help
☑ GPU Offload — n_gpu_layers: 10
0 8 16 24 max
Quick GPU Offload Settings
Low — 50/50 — Max
Detected GPU type (right click for options)
Nvidia CUDA

(b)

Inference Parameters
Parameters that control various aspects of the token prediction process
Temperature — temp: 0.8
Tokens to generate — n_predict: -1
Top K Sampling — top_k: 40
☑ Repeat penalty — repeat_penalty: 1.1
☑ Min P Sampling — min_p: 0.05
☑ Top P Sampling — top_p: 0.95
CPU Threads — n_threads: 4
Prompt Format
Message format preview

```
<|start_header_id|>system<|end_header_id|>

{System} <|eot_id|>
<|start_header_id|>user<|end_header_id|>

{User}
<|eot_id|><|start_header_id|>assistant<|end_

{Assistant}
```

(Optional) Some models require a specific format for the system prompt. If you are unsure, leave this blank.
System Message Prefix: <|start_header_id|>syste
System Message Suffix: <|eot_id|>

(c)

Figure 2. Configuration settings of LLaMA 3 for automated resume screening of (a) model selection in LM Studio, (b) GPU core count and memory allocation settings for LLaMA 3 model deployment, and (c) hardware parameters setup

2.2.4. Prompt engineering and execution

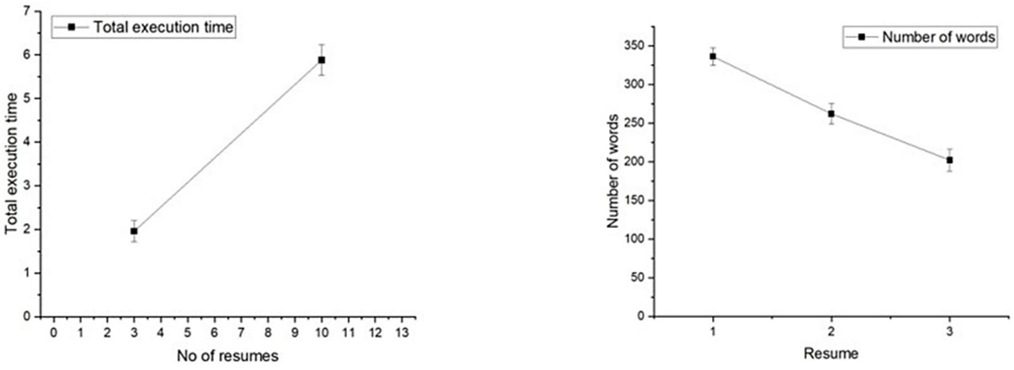
To interact with the model, specific prompts were crafted for both the job description and candidate resumes. The job description was input using the command: "I need your help in deciding among all candidates which one is the best fit for this job description: [Job Description]." Each resume was converted to structured text and concatenated into a single input file. The following prompt was used for screening: "Based on the job description and the provided candidate profiles, help me decide which candidate is the best fit for the job using the resume summary of each candidate's skills, education, achievements, and experience." LLaMA 3 processed these inputs and directly returned the name of the most suitable candidate, along with a reasoned explanation for the selection.

3. RESULTS

For this study, the LLaMA 3 model (8 B version) was deployed in LM Studio, requiring at least 8 GB GPU and more than 8 GB RAM. Key hyperparameters such as temperature (0.8) and top-k (40) were fine-tuned to balance diversity and determinism. Resume PDFs were converted to text using PyPDF2, and all resumes were evaluated using the same standardized prompt.

3.1. Execution time and batch analysis

To evaluate the scalability and processing efficiency of the proposed automated resume screening system, experiments were conducted by varying the number of resumes processed per batch. Figure 3 illustrates the system performance in terms of execution time and text volume processed. Figure 3(a) shows that a batch of three resumes took approximately 2 seconds to process, averaging 0.66 seconds per resume. As batch size increases, the total execution time shows a non-linear trend, likely due to GPU memory limitations. Figure 3(b) displays a slight decrease in the average word count per resume as the batch size increases, suggesting possible token truncation or context-length prioritization.



(a) (b)

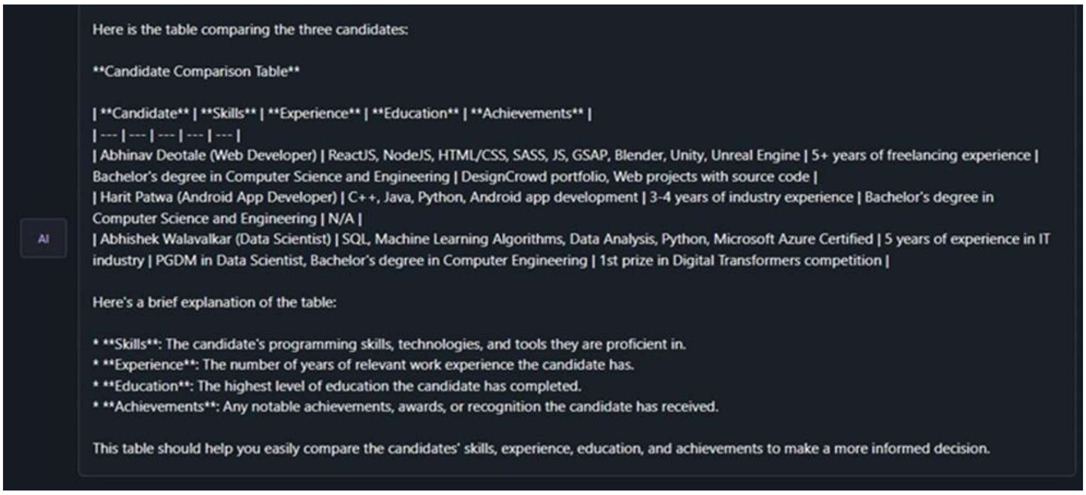
Figure 3. Performance analysis of the automated resume screening system of (a) number of resumes versus total execution time and (b) average number of words processed per resume in a batch

Analysis indicates that processing a batch of three resumes takes approximately 2 seconds, resulting in an average scanning speed of 0.66 seconds per resume. As the number of resumes increases, the processing time is expected to follow a non-linear trend and may eventually plateau due to potential GPU memory limitations. To improve processing efficiency, batch processing strategies similar to the bubble sorting algorithm can be employed. The plot in Figure 3(b) displays the average word count of resumes processed by the LLaMA 3 model for batch sizes ranging from 1 to 3. The x-axis represents the number of resumes in a batch, while the y-axis shows the average number of words per resume. The data suggests a slight decrease in average word count per resume as the batch size increases, for example with 336 words for a single resume and 202 words for a batch of three.

3.2. Comparative analysis and final decision output

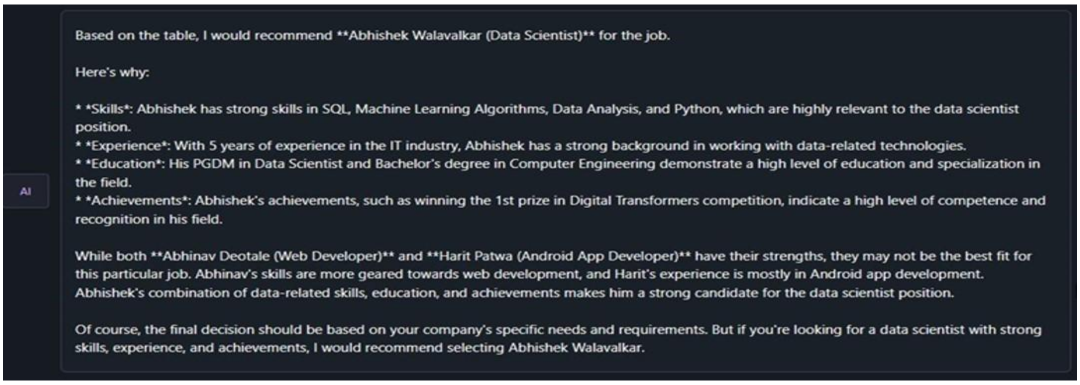
Figure 4 displays a comprehensive comparative analysis table generated by LLaMA 3, evaluating three candidates (Abhinav Deotale, Harit Patwa, and Abhishek Walawalkar) against the specified job requirements. The tabular output demonstrates the model's capability to systematically analyze and categorize candidate qualifications across multiple dimensions like, skills assessment here the output presents a detailed breakdown of each candidate's technical and soft skills, highlighting their relevance to the position requirements. Next is the experience evaluation where the model quantifies and qualifies professional experience, providing not just duration metrics but also contextual analysis of role relevance to the target position. Then stands the educational qualification analysis beyond listing degrees, the table includes an evaluation of educational relevance to the job requirements, demonstrating LLaMA 3's ability to contextualize educational backgrounds. Further is the achievement recognition being in the notable accomplishments are extracted and presented with an implicit weighting toward achievements most relevant to the position and with the last being decision support visualization which with its structured format enables rapid identification of strengths and gaps across candidates, serving as an effective decision support tool. This output represents a significant advancement over traditional keyword-based resume screening by providing standardized, multi-faceted candidate comparisons. The execution time of 2.29 seconds for generating this comprehensive analysis further demonstrates the efficiency gain over manual review processes, which typically require 15-30 minutes per resume. Hiring decisions can be affected by bias from initial information, leading recruiters to favor candidates based on early impressions rather than overall qualifications.

Figure 5, presents the proposed output generated by LLaMA 3, which identifies the candidate deemed to be the best fit for the job role. Additionally, the output provides a detailed rationale behind LLaMA 3’s decision, offering insights into the factors that make this particular candidate the most suitable for the position. This comprehensive analysis not only facilitates the selection process but also enhances transparency by providing clear justifications for the model’s decision-making. Leveraging LLaMA 3’s advanced capabilities, recruiters, and hiring managers can make more informed and objective decisions, ultimately improving the quality and efficiency of the recruitment process. It is also important to noting that this data only shows a very small sample size, so it’s not possible to draw any definitive conclusions from it. It is also unclear whether this trend would hold for larger batches of resumes.



```
Here is the table comparing the three candidates:

**Candidate Comparison Table**

| **Candidate** | **Skills** | **Experience** | **Education** | **Achievements** |
| --- | --- | --- | --- | --- |
| Abhinav Deotale (Web Developer) | ReactJS, NodeJS, HTML/CSS, SASS, JS, GSAP, Blender, Unity, Unreal Engine | 5+ years of freelancing experience | Bachelor's degree in Computer Science and Engineering | DesignCrowd portfolio, Web projects with source code |
| Harit Patwa (Android App Developer) | C++, Java, Python, Android app development | 3-4 years of industry experience | Bachelor's degree in Computer Science and Engineering | N/A |
| Abhishek Walavalkar (Data Scientist) | SQL, Machine Learning Algorithms, Data Analysis, Python, Microsoft Azure Certified | 5 years of experience in IT industry | PGDM in Data Scientist, Bachelor's degree in Computer Engineering | 1st prize in Digital Transformers competition |

Here's a brief explanation of the table:

* **Skills**: The candidate's programming skills, technologies, and tools they are proficient in.
* **Experience**: The number of years of relevant work experience the candidate has.
* **Education**: The highest level of education the candidate has completed.
* **Achievements**: Any notable achievements, awards, or recognition the candidate has received.

This table should help you easily compare the candidates' skills, experience, education, and achievements to make a more informed decision.
```

Figure 4. Comparative analysis of three candidates evaluated in a single batch using LLaMA 3



```
Based on the table, I would recommend **Abhishek Walavalkar (Data Scientist)** for the job.

Here's why:

* *Skills*: Abhishek has strong skills in SQL, Machine Learning Algorithms, Data Analysis, and Python, which are highly relevant to the data scientist position.
* *Experience*: With 5 years of experience in the IT industry, Abhishek has a strong background in working with data-related technologies.
* *Education*: His PGDM in Data Scientist and Bachelor's degree in Computer Engineering demonstrate a high level of education and specialization in the field.
* *Achievements*: Abhishek's achievements, such as winning the 1st prize in Digital Transformers competition, indicate a high level of competence and recognition in his field.

While both **Abhinav Deotale (Web Developer)** and **Harit Patwa (Android App Developer)** have their strengths, they may not be the best fit for this particular job. Abhinav's skills are more geared towards web development, and Harit's experience is mostly in Android app development. Abhishek's combination of data-related skills, education, and achievements makes him a strong candidate for the data scientist position.

Of course, the final decision should be based on your company's specific needs and requirements. But if you're looking for a data scientist with strong skills, experience, and achievements, I would recommend selecting Abhishek Walavalkar.
```

Figure 5. Final candidate selection based on job description and resume analysis using LLaMA 3

4. DISCUSSION

The proposed automated resume screening approach, leveraging the capabilities of the LLaMA 3 LLM, offers several distinct advantages over traditional methods. First, it significantly accelerates the initial screening process by rapidly analyzing large volumes of resumes, extracting relevant information, and efficiently identifying top candidates. This time-saving aspect allows recruiters and hiring managers to focus their efforts on in-depth evaluations of the most promising applicants. Moreover, LLaMA 3's contextual understanding of language enables it to move beyond simplistic keyword matching, enhancing the accuracy of candidate evaluations. By holistically analyzing resumes, the model can identify qualified individuals who may not use exact keyword phrases but possess the relevant skills and experience required for the role. The applications of this AI-driven approach extend beyond traditional job recruitment. It could be adapted for internal employee promotion processes, streamlining the identification of suitable candidates within an organization based on their skills and experience. Additionally, career counselling services could leverage this technology to match individuals with appropriate job opportunities based on their qualifications and career goals.

An experiment was conducted to compare LLaMA 3 with other leading LLMs for resume screening, with results shown in Table 1. Metrics such as accuracy, precision, and recall were used, and while LLaMA 3 showed strong results, GPT-4 achieved the highest average score. LLaMA 3, when implemented in LM Studio, supports a maximum input of 4,096 tokens (~3,000 words). Given the average resume length of 500 words, around six resumes can be processed per input. This study introduces a novel application of LLaMA 3 for automated resume screening, leveraging its contextual understanding and offline deployment through LM Studio to address privacy concerns. It presents empirical data on performance, including an average runtime of 0.66 seconds per resume in small batches, and highlights LLaMA 3's ability to provide comparative candidate analysis with explanations. GPU memory limitations and batch processing strategies are discussed, along with ethical concerns like potential model bias.

The proposed system could benefit recruitment by enabling more efficient job matching and potentially reducing unemployment. Its objective assessments may help minimize human bias, fostering inclusivity, and fairness. By automating initial screening, recruiters can focus on in-depth evaluation and cultural fit. The model also has potential applications in career counselling and internal promotions. However, its effectiveness depends on diverse, unbiased training data. While not a new algorithm, this work integrates LLaMA 3 into a complete workflow involving PDF-to-text conversion, job description prompting, and batch optimization within LM Studio. The system should support, not replace, human judgment. Despite limitations, LLaMA 3 shows significant promise in modernizing and improving recruitment practices.

Table 1. Comparison of resume screening performance using different LLMs

Model	Average score
LLaMA 3	66.92
GPT-4	73.85
Claude 3 Sonnet	64.66
Gemini Pro 1.5	69.10

5. CONCLUSION

This study highlights the significant potential of LLMs, particularly Meta AI's LLaMA 3, in transforming automated resume screening systems. levant skills and experiences and matching candidates to job descriptions more accurately than traditional keyword-based methods. By emplfoying LLaMA 3's advanced natural language understanding capabilities, the proposed system surpasses traditional keyword-based approaches through contextual analysis of candidate profiles. Unlike earlier systems that rely heavily on surface-level term matching, LLaMA 3 effectively interprets the semantic relevance of a candidate's skills, qualifications, and experience in relation to specific job descriptions. The system demonstrated an average scanning speed of 0.66 seconds per resume for small batches, though larger batches may require optimized processing due to GPU constraints. LLaMA 3 also outperformed basic keyword approaches by evaluating candidate qualifications holistically. However, scalability challenges were noted as batch size increased, primarily due to GPU memory and processing constraints, indicating the need for model optimization or hardware augmentation in large-scale deployments. Future work should enhance interpretability and fairness in LLaMA 3-based screening through explainable AI and bias mitigation techniques. Also improving dataset diversity and designing effective prompts can boost accuracy and reduce unfair outcomes. Domain specific fine tuning and human in the loop feedback will support continuous model refinement.

FUNDING INFORMATION

The author states that no funding was involved in this project.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Asmita Deshmukh	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Anjali Raut		✓				✓				✓	✓	✓	✓	
Vedant Deshmukh		✓	✓	✓			✓			✓	✓			

C :	**C**onceptualization	I :	**I**nvestigation	Vi :	**Vi**sualization
M :	**M**ethodology	R :	**R**esources	Su :	**Su**pervision
So :	**So**ftware	D :	**D**ata Curation	P :	**P**roject administration
Va :	**Va**lidation	O :	Writing - **O**riginal Draft	Fu :	**Fu**nding acquisition
Fo :	**Fo**rmal analysis	E :	Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Specific data required can be made available upon reasonable request to the corresponding author, [AD]. All materials belong to the authors and cannot be sold to anyone. Code will be made available upon reasonable request to the corresponding author.

REFERENCES

[1] S. K. Kopparapu, "Automatic extraction of usable information from unstructured resumes to aid search," in *2010 IEEE International Conference on Progress in Informatics and Computing*, Dec. 2010, pp. 99–103, doi: 10.1109/PIC.2010.5687428.

[2] A. B. Chapman *et al.*, "Using natural language processing to study homelessness longitudinally with electronic health record data subject to irregular observations," Mar. 18, 2023, doi: 10.1101/2023.03.17.23287414.

[3] D. J. Feller, J. Zucker, M. T. Yin, P. Gordon, and N. Elhadad, "Using clinical notes and natural language processing for automated HIV risk assessment," *JAIDS Journal of Acquired Immune Deficiency Syndromes*, vol. 77, no. 2, pp. 160–166, Feb. 2018, doi: 10.1097/QAI.0000000000001580.

[4] J. Nayor, L. F. Borges, S. Goryachev, V. S. Gainer, and J. R. Saltzman, "Natural language processing accurately calculates adenoma and sessile serrated polyp detection rates," *Digestive Diseases and Sciences*, vol. 63, no. 7, pp. 1794–1800, Jul. 2018, doi: 10.1007/s10620-018-5078-4.

[5] H. M. Trivedi *et al.*, "Large scale semi-automated labeling of routine free-text clinical records for deep learning," *Journal of Digital Imaging*, vol. 32, no. 1, pp. 30–37, Feb. 2019, doi: 10.1007/s10278-018-0105-8.

[6] B. Gunaseelan, S. Mandal, and V. Rajagopalan, "Automatic extraction of segments from resumes using machine learning," in *2020 IEEE 17th India Council International Conference (INDICON)*, Dec. 2020, pp. 1–6, doi: 10.1109/INDICON49873.2020.9342596.

[7] K. Wailthare, A. Tamhane, V. Mulik, and K. Suryawansh, "A cosine similarity-based resume screening system for job recruitment," *International Research Journal of Modernization in Engineering Technology and Science*, Apr. 2023, doi: 10.56726/IRJMETS35945.

[8] P. K. Roy, S. S. Chowdhary, and R. Bhatia, "A machine learning approach for automation of resume recommendation system," *Procedia Computer Science*, vol. 167, pp. 2318–2327, 2020, doi: 10.1016/j.procs.2020.03.284.

[9] A. H. Alderham and E. S. Jaha, "Improved candidate-career matching using comparative semantic resume analysis," *Advances in Science, Technology and Engineering Systems Journal*, vol. 9, no. 1, pp. 15–22, Jan. 2024, doi: 10.25046/aj090103.

[10] N. Sharma, R. Bhutia, V. Sardar, A. P. George, and F. Ahmed, "Novel hiring process using machine learning and natural language processing," in *2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Jul. 2021, pp. 1–6, doi: 10.1109/CONECCT52877.2021.9622692.

[11] A. Dossatayev, A. Manapova, and B. Omarov, "Deep residual convolutional long short-term memory network for option price prediction problem," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, 2023, doi: 10.14569/IJACSA.2023.0140941.

[12] S. Bharadwaj, R. Varun, P. S. Aditya, M. Nikhil, and G. C. Babu, "Resume screening using NLP and LSTM," in *2022 International Conference on Inventive Computation Technologies (ICICT)*, Jul. 2022, pp. 238–241, doi: 10.1109/ICICT54344.2022.9850889.

[13] S. L. Oh, E. Y. K. Ng, R. S. Tan, and U. R. Acharya, "Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats," *Computers in Biology and Medicine*, vol. 102, pp. 278–287, Nov. 2018, doi: 10.1016/j.compbiomed.2018.06.002.

[14] A. Sharma, N. Garg, S. Patidar, R. San Tan, and U. R. Acharya, "Automated pre-screening of arrhythmia using hybrid combination of Fourier–Bessel expansion and LSTM," *Computers in Biology and Medicine*, vol. 120, May 2020, doi: 10.1016/j.compbiomed.2020.103753.

[15] C. Athukorala, H. Kumarasinghe, K. Dabare, P. Ujithangana, S. Thelijjagoda, and P. Liyanage, "Business intelligence assistant for human resource management for IT companies," in *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, Nov. 2020, pp. 220–225, doi: 10.1109/ICTer51097.2020.9325471.

[16] G. L, R. M L, G. H B, K. Mathada, and M. B, "Intelligent resume scrutiny using named entity recognition with BERT," in *2023 International Conference on Data Science and Network Security (ICDSNS)*, Jul. 2023, pp. 01–08, doi: 10.1109/ICDSNS58469.2023.10245304.

[17] E. Abdollahnejad, M. Kalman, and B. H. Far, "A deep learning BERT-based approach to person-job fit in talent recruitment," in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec. 2021, pp. 98–104, doi: 10.1109/CSCI54926.2021.00091.

[18] M. B. Padmaja, C. S. Kumar, V. Preethi, B. S. Someswar, and A. Gantayat, "A web application-based mock interview using NLP techniques," *International Journal of Advanced Research in Education and Technology*, vol. 11, no. 6, pp. 2669–2677, 2024, doi: 10.15680/IJARETY.2024.1106016.

[19] S. A. Pias, M. Hossain, H. Rahman, and Md. M. Hossain, "Enhancing job matching through natural language processing: a BERT-based approach," in *2024 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, Oct. 2024, pp. 1–6, doi: 10.1109/ICISET62123.2024.10939860.

[20] V. James, A. Kulkarni, and R. Agarwal, "Resume shortlisting and ranking with transformers," in *Intelligent Systems and Machine Learning*, vol. 471, S. Nandan Mohanty, V. Garcia Diaz, and G. A. E. Satish Kumar, Eds., Cham: Springer Nature Switzerland, 2023, pp. 99–108, doi: 10.1007/978-3-031-35081-8_8.

[21] M. Kaya and T. Bogers, "An exploration of sentence-pair classification for algorithmic recruiting," in *Proceedings of the 17th ACM Conference on Recommender Systems*, Sep. 2023, pp. 1175–1179, doi: 10.1145/3604915.3610657.

[22] M. Li, J. Sun, and X. Tan, "Evaluating the effectiveness of large language models in abstract screening: a comparative analysis," *Systematic Reviews*, vol. 13, no. 1, Aug. 2024, doi: 10.1186/s13643-024-02609-x.

[23] C. Gan, Q. Zhang, and T. Mori, "Application of LLM agents in recruitment: a novel framework for automated resume screening," *Journal of Information Processing*, vol. 32, pp. 881–893, 2024, doi: 10.2197/ipsjjip.32.881.

[24] P. Patil, A. Kharde, S. Deochake, and V. Kharat, "Application of RAG (retrieval-augmented generation) in AI-driven resume analysis and job matching," *International Journal of Advanced Research in Science, Communication and Technology*, pp. 671–679, Mar. 2025, doi: 10.48175/IJARSCT-25189.

[25] F. P.-W. Lo *et al.*, "AI hiring with LLMs: a context-aware and explainable multi-agent framework for resume screening," 2025, *arXiv:2504.02870*.

BIOGRAPHIES OF AUTHORS

Asmita Deshmukh is assistant professor at KJSIT, Mumbai, India. She is currently pursuing Ph.D. (Computer Science Engineering) from Sant Gadge Baba Amravati University, master of engineering (Computer Engineering) from University of Mumbai and bachelor of engineering (Computer Science Engineering) from Amravati University in 1996 and 2012. Her primary research interests include natural language processing and deep learning. She can be contacted at email: asmitadeshmukh7@gmail.com.

Anjali Raut is professor and incharge principal at HVPM College of Engineering and Technology, Amravati India. She has completed her Ph.D. (Computer Science Engineering) from Sant Gadge Baba Amravati University, master of engineering (Computer Science Engineering) and bachelor of engineering (Computer Science Engineering) from Amravati University in 1994 and 2013. Her main area of research interests is data mining. She can be contacted at email: anjali_dahake@rediffmail.com.

Vedant Deshmukh is a final-year student at the Department of Computer Engineering, Sardar Patel Institute of Technology (SPIT), Mumbai, India. He is expected to receive his B.Tech. degree in Computer Engineering in 2025. His research interests include medical imaging using computer vision and machine learning. He can be contacted at email: vedantdeshmukh3108@gmail.com.