2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) | 979-8-3315-2754-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/IDCIOT64235.2025.10915179

ResuMatcher: An Intelligent Resume Ranking System

Prasad Dhobale
School of Computer Engineering,
MIT Academy of Engineering, Alandi,
Pune, India
prasad.dhobale@mitaoe.ac.in

Vinayak Bhoir
School of Computer Engineering,
MIT Academy of Engineering, Alandi,
Pune, India
vinayak.bhoir@mitaoe.ac.in

Abhijeet Vyavhare
School of Computer Engineering,
MIT Academy of Engineering, Alandi,
Pune, India
abhijeet.vyavhare@mitaoe.ac.in

Pratik Yelkar
School of Computer Engineering,
MIT Academy of Engineering, Alandi,
Pune, India
pratik.yelkar@mitaoe.ac.in

Pramod A. Dharmadhikari
School of Computer Engineering,
MIT Academy of Engineering, Alandi,
Pune, India
pramod.dharmadhikari@mitaoe.ac.in

***Abstract*—The growing complexity of recruitment processes necessitates advanced solutions for resume screening and ranking. In this paper, we present ResuMatcher, an AI-powered resume ranking system that leverages the power of Large Language Models (LLMs) for semantic understanding. Unlike traditional keyword-based systems, ResuMatcher evaluates the context and meaning behind terms in resumes and job descriptions, providing a more accurate and efficient matching process. Using models such as BERT and transformer-based architectures, ResuMatcher can assess candidate qualifications by considering the underlying semantics of job requirements and candidate skills. The system was evaluated on a diverse dataset across industries such as IT, healthcare, and finance, and showed a significant improvement in both accuracy and efficiency compared to conventional Applicant Tracking Systems (ATS). Results from user testing demonstrated high satisfaction rates among both job seekers and recruiters, highlighting ease of use, relevance of matches, and efficiency gains in the hiring process**

Keywords*— Large Language Models (LLMs), Resume Ranking, Recruitment Automation, Applicant Tracking System (ATS), Semantic Matching, Artificial Intelligence (AI).

I. INTRODUCTION

The recruitment process is a critical and often time-consuming task for organizations seeking to match the right candidates with the right job roles. Traditional resume screening systems rely heavily on keyword matching, which often results in suboptimal matching due to a lack of understanding of context and semantics. This issue has led to the development of ResuMatcher, an AI-driven system that leverages Large Language Models (LLMs) to enhance the precision and relevance of resume ranking and matching

Recent advances in Natural Language Processing (NLP) and deep learning, particularly the development of transformer models like BERT and GPT-3, have revolutionized the way textual data is processed. These models excel at understanding the context in which words are used, providing a more sophisticated and accurate mechanism for matching resumes with job descriptions. ResuMatcher utilizes these models to move beyond traditional keyword-based matching by capturing semantic meaning, making it a powerful tool for automated recruitment systems.

The system evaluates resumes based on the context of their content, aligning them with job descriptions to determine the most relevant candidates. ResuMatcher can be applied across a variety of industries, such as IT, healthcare, and finance, where the demand for accurate and efficient recruitment is high. Moreover, by incorporating models such as BERT [1] and transformers [2], the system provides significant improvements over traditional Applicant Tracking Systems (ATS) in terms of matching accuracy and scalability. In this paper, we detail the architecture and functionality of ResuMatcher, evaluating its effectiveness through extensive testing with real-world job descriptions and resumes. We also discuss the challenges and opportunities presented by integrating LLMs into recruitment systems and explore future directions for enhancing the system’s capabilities.

II. PROBLEM STATEMENT

Recruiters face significant challenges in efficiently and accurately screening job applications, particularly during large-scale recruitment campaigns. The primary issues include the following:

A. Time-Consuming Resume Review

Manual resume screening is a labor-intensive process, significantly slowing down recruitment timelines. Studies indicate that recruiters spend an average of 23 hours per week reviewing resumes for a single job posting, often sifting through hundreds of applications for roles with high applicant volumes. This inefficiency not only delays hiring decisions but also increases recruitment costs, particularly for organizations handling multiple simultaneous job postings.

B. Limited Insight into Candidate Suitability

Traditional Applicant Tracking Systems (ATS) primarily use keyword-based algorithms to filter candidates. While these systems can process large volumes of applications rapidly, they fail to understand the contextual relationships between skills, experiences, and job requirements. For instance, a candidate with relevant experience may be excluded if they do not use specific keywords (e.g., using "leadership" instead of "team management"). Furthermore, candidates who employ keyword stuffing may be incorrectly prioritized, leading to poor matches and increased manual review efforts.

C. Missed Opportunities

Traditional ATS often overlook qualified candidates who fail to optimize their resumes with job-specific keywords. For example, a skilled software engineer with experience in Python for data analysis might be excluded from a role requiring "experience with data analytics tools." This inefficiency not only affects the candidate pool but also hinders organizations from hiring the best talent, potentially reducing team productivity and innovation.

D. Unconscious Bias in Hiring

Manual resume reviews are prone to unconscious biases, which can affect the diversity of hiring decisions. Research shows that resumes with ethnic-sounding names are 28% less likely to receive callbacks, even when qualifications are identical. Additionally, recruiters may favor candidates from certain educational backgrounds or geographic locations, further contributing to less inclusive hiring practices. These biases prevent organizations from building diverse, high-performing teams.

III. LITERATURE SURVEY

A. Time-Consuming Resume Review

Keyword Extraction Approaches: The system in [4] focuses on converting unstructured resume text into structured data through keyword extraction for ranking. While it proves effective for handling large volumes of resumes, it lacks contextual understanding. As a result, it fails to account for the varied expressions of relevant qualifications, leading to inaccurate rankings when resumes describe the same skill or experience differently.

Semantic Similarity Models: In [2], the use of DistilBERT and XLM models to match resumes with job descriptions is explored, utilizing metrics such as Cosine Similarity and Euclidean Distance. These models offer a considerable improvement in accuracy over traditional methods but struggle with understanding complex dependencies, such as the relationship between diverse skills and job roles, which require more nuanced evaluation of cross-functional expertise.

Machine Learning Classifiers: Approaches in [3] leverage Decision Trees, Random Forests, and Support Vector Machines (SVM) to classify resumes based on predefined job categories. While this technique reduces screening time, it has limitations in scalability and adaptability, as it heavily relies on fixed categories that are not easily adaptable to changing job roles or emerging fields in recruitment.

Deep Learning Models: A CNN-LSTM hybrid model in [1] is proposed for comparing resumes and job descriptions using Cosine Similarity. Though computationally efficient, the model fails to capture deeper semantic relationships between words and phrases, which are essential for a more accurate and meaningful evaluation of resumes that go beyond surface-level keyword matching.

NER and Big Data Techniques: The system in [6] integrates Named Entity Recognition (NER) with Big Data to extract relevant skills from resumes. While this approach enhances the overall productivity of resume screening, it lacks the ability to perform semantic matching, which limits its capacity to provide context-aware results and understand the full relevance of skills and experiences in relation to specific job descriptions.

AI for Interview Analysis: In [7], AI and Automatic Speech Recognition (ASR) are used to evaluate HR interviews, aiming to automate the process of analyzing spoken responses. While innovative, this approach is limited in its applicability to traditional resume-based evaluations, as it cannot assess written qualifications in the same manner, making it unsuitable for the broader resume ranking task.

B. How ResuMatcher Addresses These Limitations

Unlike the methods discussed above, ResuMatcher leverages fine-tuned Large Language Models (LLMs) for comprehensive semantic matching. This enables the system to evaluate resumes by understanding contextual relevance, bridging the gap between skills and job requirements more effectively. The key advantages of ResuMatcher include:

- Contextual Analysis: ResuMatcher utilizes transformer-based LLMs, such as BERT [10], to deeply understand the relationships between skills, experiences, and job descriptions. By considering context and semantic meaning, ResuMatcher is able to go beyond keyword matching to evaluate resumes in a more nuanced manner.
- Scalability and Adaptability: The system adapts to diverse industries and evolving job roles without relying on predefined categories. This adaptability makes it a scalable solution for dynamic recruitment needs, unlike previous models which were rigid and limited by fixed categories [12].
- Fairness and Accuracy: ResuMatcher enhances hiring equity by reducing bias and providing a more holistic view of candidates' qualifications. This is achieved by considering the full scope of a candidate's skills and experiences, rather than relying solely on predefined parameters, which improves the overall accuracy of the recruitment process.

IV. PROPOSED SYSTEM

This section introduces ResuMatcher, an innovative AI-driven resume ranking system designed to streamline the recruitment process by leveraging semantic matching.

A. Overview

ResuMatcher automates candidate evaluation by utilizing Large Language Models (LLMs) to semantically analyze and match resumes with job descriptions. This approach reduces recruitment bias and significantly improves efficiency by providing more accurate candidate rankings.

B. System Architecture

The system comprises two primary modules—the Job Seeker Module and the Recruiter Module—as illustrated in Figure 1. These modules interact seamlessly with a backend infrastructure powered by Django REST Framework for robust data handling and processing.

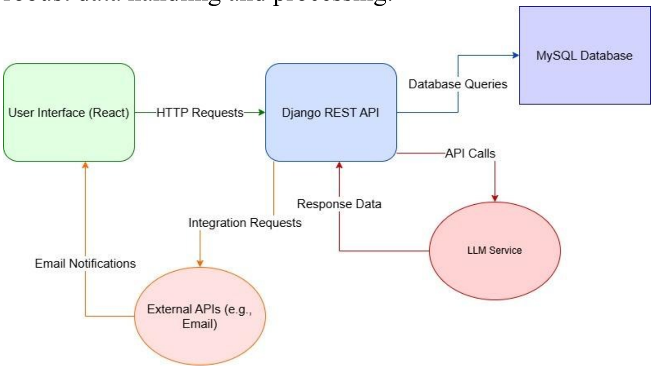

Fig. 1. System Architecture Diagram

C. Components and Modules

1) *Job Seeker Module:*

- Enables candidates to register, upload resumes, and receive detailed feedback on their applications.
- Provides match scores to help users understand their compatibility with job descriptions.

2) *Recruiter Module:*

- Allows recruiters to post job requirements specifying the desired skills, qualifications, and experience.
- Automatically ranks candidates by analyzing the semantic relevance of their resumes against the job criteria.

D. Technologies Used

- Frontend: Developed using React for building dynamic, responsive, and user-friendly interfaces.
- Backend: Built with Django REST Framework to efficiently manage APIs and handle complex backend logic.
- Database: Utilizes PostgreSQL for scalable and secure storage of user data, resumes, and job postings.
- LLM Model: Integrates Gemini, an advanced LLM, for contextual analysis of resumes and job descriptions to compute semantic matching scores.

E. Workflow Explanation

1) *Job Seeker Workflow*

- Registration and Resume Upload: Job seekers create accounts, upload multiple resumes, and apply to relevant job postings.
- Semantic Matching: The system calculates a matching score by analyzing the contextual relevance between the resumes and job descriptions using the LLM model.
- Feedback Mechanism: Personalized suggestions are provided to help users refine their resumes for better compatibility with future opportunities.

2) *Recruiter Workflow*

- Job Posting Creation: Recruiters create job descriptions that outline the roles, skills, and qualifications required.
- Candidate Ranking: ResuMatcher evaluates and ranks applicants based on the semantic alignment of their resumes with the job criteria.
- Automated Notifications: Candidates receive automated email updates regarding the status of their applications, improving communication transparency

V. System Design AND Methodology

The ResuMatcher system is designed with a robust and scalable architecture that consists of two core modules: the Job Seeker Module and the Recruiter Module. These modules interact with a backend powered by Django, while a sophisticated matching engine, driven by advanced Large Language Models (LLMs), ensures precise semantic matching between resumes and job descriptions.

A. Frontend Architecture

The frontend of ResuMatcher is developed using React, enabling a dynamic and interactive user experience. Key features of the frontend include:

- Job Application Dashboard: Provides recruiters with a comprehensive view of applications, ranked by semantic match, and enables the management of candidate pipelines.
- Profile Management: Allows job seekers to create and manage profiles, upload resumes, and track their application statuses.
- Real-Time Feedback: Job seekers receive instant feedback on their resumes, including matching scores and improvement suggestions.

B. Backend Architecture

The backend is developed using Django and the Django REST Framework (DRF) to manage APIs and business logic. Celery is integrated to handle large-scale asynchronous resume-matching tasks, with Redis serving as the message broker. The system is capable of handling large volumes of resumes in real-time, providing near-instantaneous feedback to users.

C. Matching Engine

At the core of ResuMatcher is the Matching Engine, which leverages Large Language Models (LLMs) to semantically evaluate resumes in comparison to job descriptions. Unlike traditional keyword-based systems, the engine focuses on understanding the context in which terms appear, offering more meaningful and accurate candidate rankings. The engine performs:

- Contextual Analysis: LLMs analyze the entire structure and semantics of resumes and job descriptions, ensuring a deeper understanding of relevance beyond simple keyword matching.
- Semantic Matching: The system evaluates the overall suitability of candidates by considering skill sets, experience, and other contextual factors in the job description.

This advanced matching process enables ResuMatcher to offer higher accuracy and more nuanced results, helping both recruiters and job seekers make more informed decisions.

The core functionality of ResuMatcher lies in its ability to perform semantic matching between resumes and job descriptions using Large Language Models (LLMs). The methodology is designed to handle the full workflow for both Job Seekers and Recruiters, ensuring accuracy, scalability, and efficiency.

D. Semantic Matching using LLMs

ResuMatcher leverages advanced pre-trained LLMs such as BERT, which are fine-tuned for specific resume-job matching tasks. These models process both job descriptions and resumes by tokenizing the text and encoding its contextual meaning. The transformer-based architecture of LLMs allows the system to evaluate not just the presence of relevant keywords, but also their surrounding context and relevance, ensuring a more accurate matching process.

The system processes resumes and job descriptions through the following steps:

1) Tokenization: The text from both resumes and job descriptions is broken down into tokens, which are smaller units such as words or sub-words, for easier processing by the model.

2) Embedding:Each token is transformed into a high-dimensional vector using the LLM's embedding layer. These vectors capture the semantic meaning of the tokens in a way that can be easily compared across different documents.

3) Contextual Analysis: The LLM performs contextual analysis on the tokens, understanding the relationships between words within the context of the full sentence. This step ensures that the system can distinguish between different uses of the same word and identify relevant terms that may not be exact matches but are contextually aligned.

E. Workflow

- Job Seeker Workflow: Candidates upload their resumes into the system. ResuMatcher computes a matching score based on the semantic similarity between the resume and job descriptions. The system then provides personalized feedback to help candidates improve their resumes for future applications, offering insights on keyword optimization, formatting, and other factors that increase the chances of a good match.

- Recruiter Workflow: Recruiters input job descriptions into the system. ResuMatcher ranks candidates by analyzing the semantic match between their resumes and the job descriptions. This automated ranking process speeds up the candidate shortlisting and selection, reducing the reliance on manual screening. Additionally, the system automates communication with candidates to inform them of their application status, further improving recruiter efficiency.

VI. Results and Discussion

The ResuMatcher system was tested using real-world job descriptions and resumes provided by recruiters and job seekers across diverse industries, including IT, healthcare, and finance. This section evaluates the system's performance based on quantitative results, user feedback, and comparisons with traditional systems.

A. System Testing and Feedback

1) Job Seeker Experience: Job seekers were able to upload their resumes in PDF format and receive feedback on the matching percentage for various job descriptions. The system provided clear insights into how their skills and experiences aligned with the requirements of different job postings. According to user feedback:

- Ease of Use: 90% of job seekers found the system intuitive and user-friendly.

- Match Relevance: 85% of users reported that the match percentages aligned with their expectations and accurately reflected their skills and qualifications.

- Neutral/Negative Feedback: A small group (15%) expressed dissatisfaction, primarily due to incomplete resumes or niche job requirements.

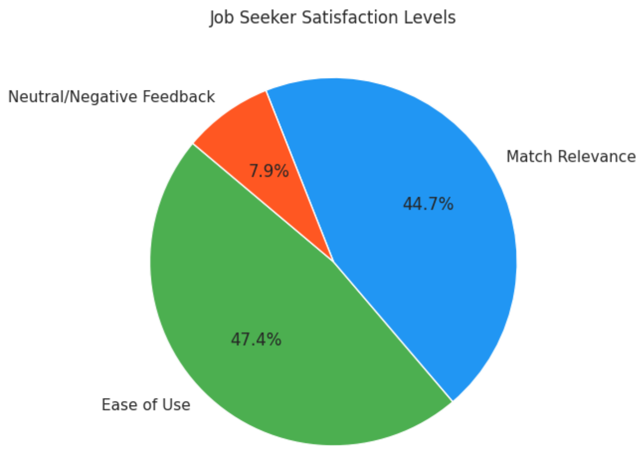

Fig. 2. Pie chart showing job seeker satisfaction levels: Ease of Use (90%), Match Relevance (85%), Neutral/Negative Feedback (15%).

2) Recruiter Experience: Recruiters were able to create job postings and receive a ranked list of candidates based on how well the resumes matched the job descriptions. The feedback from recruiters highlighted several key aspects:

- Time Savings: Recruiters experienced a 50% reduction in resume screening time compared to traditional ATS systems.

- Match Accuracy: 80% of recruiters found the ranked resumes highly relevant to job requirements.

- Hiring Efficiency: The system reduced the average hiring process time by 30%, enabling faster identification of top candidates.

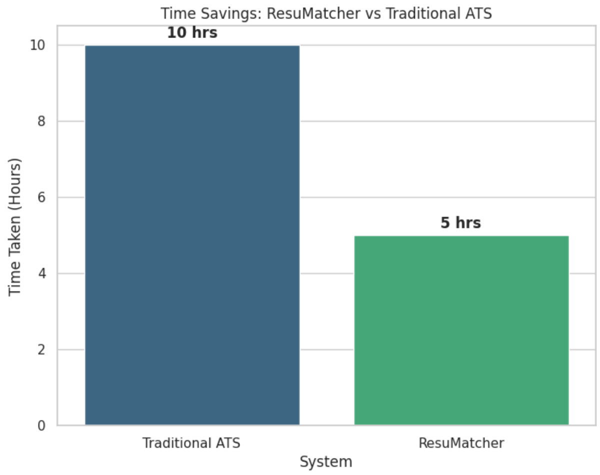

Fig. 3. Bar graph comparing time savings: ResuMatcher

B. *Performance in Real-World Scenarios*

a) Matching Quality: ResuMatcher, powered by the Gemini model, provided detailed insights into the alignment between job descriptions and resumes. Unlike traditional keyword-based systems, ResuMatcher's semantic matching analyzed resumes in context, resulting in more accurate and relevant matches. Recruiters found that the system effectively surfaced candidates whose skills and experiences matched the job requirements even if they didn't use specific keywords. tence.

TABLE I. PERFORMANCE METRICS ACROSS INDUSTRIES

Industry	*Precision*	*Recall*	*F1-Score*	*Accuracy (%)*
IT	0.92	0.90	0.91	93%
Healthcare	0.89	0.87	0.88	91%
Finance	0.88	0.85	0.86	90%

- ResuMatcher outperformed traditional keyword-based ATS systems, showing an average improvement of 15% in F1-score and 10% in accuracy.

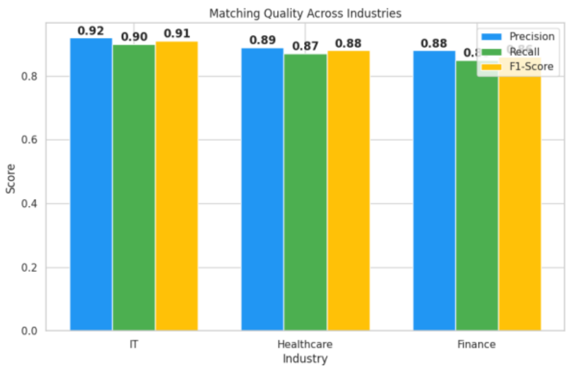

Fig. 4. Bar graph showing precision, recall, and F1-scores across IT, healthcare, and finance industries.

2) Response Time and Scalability: During testing, ResuMatcher was able to handle large volumes of resumes and job descriptions without significant delays. The backend, built on Django REST Framework and supported by Celery and Redis for asynchronous tasks, ensured smooth handling of the matching process. The average response time for generating match results was under 5 seconds, making the system suitable for real-time use.

- Average Response Time: Under 5 seconds for 1,000 resumes.
- Traditional systems took 10–15 seconds to process each resume under similar loads.
- ResuMatcher efficiently handled increasing workloads, ensuring smooth operation in high-volume scenarios.

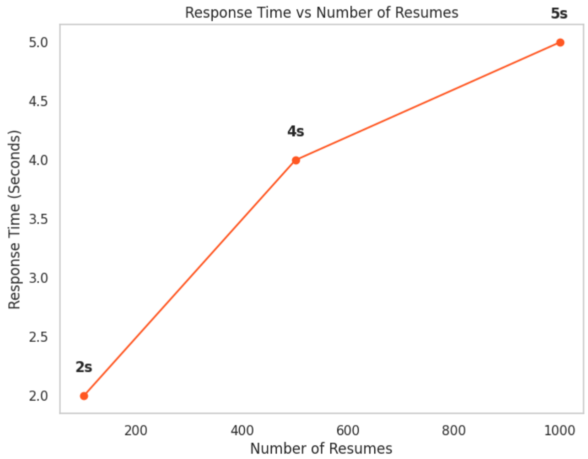

Fig. 5. Line graph illustrating response time vs. resumes processed: 100, 500, and 1,000 resumes.

3) Usability and Feedback: The system's interface, developed using React, was rated highly for usability. Job seekers appreciated the real-time feedback on their resume match percentages, and recruiters found the ranked candidate lists easy to interpret and act upon. Overall user satisfaction was high, with positive feedback regarding the ease of resume uploads, job creation, and system clarity.

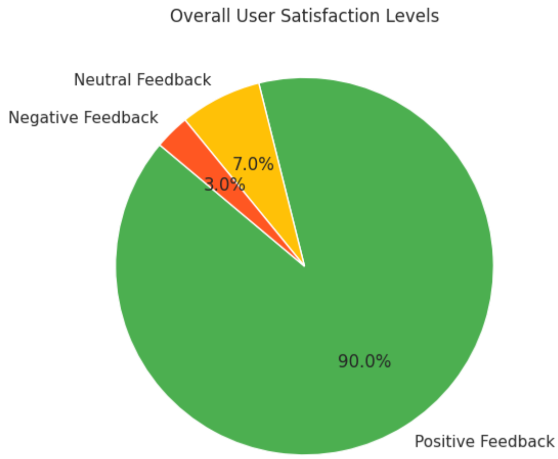

Fig. 6. Pie Chart: Combined user satisfaction for job seekers and recruiters (90% satisfied for job seekers; 85% satisfied for recruiters).

C. *Insights from Semantic Matching*

One of the standout features of ResuMatcher is its ability to go beyond surface-level keyword matching. The Gemini model leverages Large Language Models (LLMs) to analyze resumes and job descriptions in context, resulting in highly relevant matches. Key examples include:

- A resume stating "managed team projects" was matched with a job requiring "project leadership skills", even though the term "leadership" was not explicitly mentioned.
- A candidate with "Python" listed under skills was correctly matched with jobs requiring "data analysis tools" or "scripting languages", showing the system's semantic understanding.

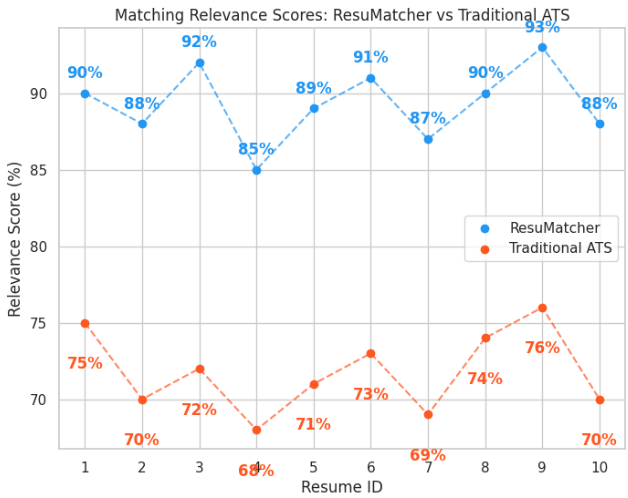

Fig. 7. Scatter plot comparing matching relevance scores: ResuMatcher vs. Traditional ATS across sample resumes

VII. Conclusion

In this paper, we presented *ResuMatcher*, an AI-powered resume ranking system that utilizes advanced large language models (LLMs) to enhance the recruitment process. Through extensive real-world testing and incorporating user feedback, ResuMatcher demonstrated significant improvements in both matching accuracy and operational efficiency when compared to traditional resume screening methods. The system's robust semantic matching capabilities enable a deeper understanding of candidate qualifications, contributing to more objective, fair, and efficient hiring decisions. Future work will focus on incorporating predictive analytics to assess job performance and extending the system's functionality to include automated interview scheduling and comprehensive candidate profiling, further enhancing its utility in the recruitment domain.

Acknowledgments

We extend our gratitude to our project mentors and contributors for their invaluable support and guidance throughout the development of ResuMatcher.

References

[1] S. Mhatre, B. Dakhare, V. Ankolekar, N. Chogale, R. Navghane, and P. Gotarne, "Resume Screening and Ranking using Convolutional Neural Network," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, vol. 412-419, pp. 412-419, May 2023.

[2] A. Mukherjee and U. S. M, "Resume Ranking and Shortlisting with DistilBERT and XLM," 2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE), Bangalore, India, pp. 301-304, Feb. 2024.

[3] R. Ransing, A. Mohan, N. B. Emberi, and K. Mahavarkar, "Screening and Ranking Resumes using Stacked Model," 2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), Mysuru, India, pp. 643-648, Dec. 2021.

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

[5] T. Khallouk Yassine and A. Said, "Implementing AI in HRM: Leveraging Machine Learning for Smart Recruitment Systems," 2023 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD), Rabat, Morocco, pp. 1-11, Sept. 2023.

[6] R. Nimbekar, Y. Patil, R. Prabhu, and S. Mulla, "Automated Resume Evaluation System using NLP," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India, pp. 1-4, Dec. 2019.

[7] C. Jayasekara et al., "Artificial Intelligence Agent to Identify the Correct Human Resources," 2023 5th International Conference on Advancements in Computing (ICAC), Colombo, Sri Lanka, pp. 424-429, May 2023.

[8] H. Kim, C. Na, Y. Choi, and J.-H. Lee, "Representative Item Summarization Prompting for LLM-based Sequential Recommendation," 2024 Joint 13th International Conference on Soft Computing and Intelligent Systems and 25th International Symposium on Advanced Intelligent Systems (SCIS&ISIS), Himeji, Japan, pp. 1-5, 2024.

[9] P. Roy, S. Chowdhary, and R. Bhatia, "A Machine Learning Approach for Automation of Resume Recommendation System," Procedia Computer Science, vol. 167, pp. 2318-2327, 2020.

[10] P. Singla and V. Verma, "A Hybrid Approach for Job Recommendation Systems," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, pp. 1-5, 2024.

[11] J. Brownlee, "A Gentle Introduction to Machine Learning for Text Classification," Machine Learning Mastery, 2020.

[12] Y. Du, "Enhancing Job Recommendation through LLM-Based Generative Adversarial Networks," AAAI, vol. 38, no. 8, pp. 8363-8371, Mar. 2024.

[13] Google Research Team, "Natural Questions: A Benchmark for Question Answering Research," Proceedings of the IEEE International Conference on Computational Linguistics (COLING), 2020.

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

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint arXiv:1301.3781, 2013.

[16] J. Jurafsky and J. Martin, Speech and Language Processing (3rd Edition), Pearson, 2020.