AI-based Multimodal Resume Ranking Web Application for Large Scale Job Recruitment

Mehmet Batuhan Yazıcı
Computer Engineering Department
Istanbul Kultur University
Istanbul, Turkey
2000003436@stu.iku.edu.tr

Damla Sabaz
Computer Engineering Department
Istanbul Kultur University
Istanbul, Turkey
1900005131@stu.iku.edu.tr

Wisam Elmasry
Computer Engineering Department
Istanbul Kultur University
Istanbul, Turkey
w.elmasry@iku.edu.tr

2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP) | 979-8-3315-3149-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/IDAP64064.2024.10710945

***Abstract*—This paper presents a resume-ranking web application that improves recruitment through advanced deep-learning techniques. The system uses the YOLOv9 model fine-tuned with our newly created custom dataset for segment detection on resumes of various structures, EasyOCR for text recognition, mBERT fine-tuned for text classification, and GLiNER for named entity recognition with regular expressions. These models and techniques efficiently extract, categorize, and match resume information with job descriptions. We created a custom dataset for our object detection training, and while we trained three models, YOLOv9 achieved the highest performance with a score of 0.84 mAP. Our hybrid matching approach provides highly accurate and relevant resume rankings using the embedding model, gte-large-en-v1.5, and cosine similarity for semantic matching with dense vectors with extracted keywords and BM25 for keyword relevance. The web application allows HR professionals to upload resumes seamlessly, define job descriptions, and view ranked results, providing a tailored solution to specific recruitment needs. Although we faced challenges such as text extraction accuracy and zero-shot NER limitations, our system demonstrated a solid overall performance. This paper demonstrates the potential of state-of-the-art deep learning models to enhance recruitment processes and provides a valuable tool for HR professionals to identify the most suitable candidates efficiently.**

***Index Terms*—Artificial Intelligence, Natural Language Processing, Optical Character Recognition, Named Entity Recognition, Text Recognition**

I. Introduction

The growth of online job platforms and remote work opportunities has led to a surge in job applications, putting pressure on Human Resources (HR) departments. This paper introduces a web application that uses deep learning techniques to help HR departments rank job applicants more effectively. This solution allows HR teams to tackle the growing challenges in the hiring process [1] [2]. These teams face considerable obstacles while accurately aligning resumes with job descriptions. The traditional method of manually sorting through resumes is slow, error-prone, and often results in the oversight of qualified candidates [3]. Human reviewers can only process a limited number of resumes daily. Furthermore, human biases can affect decision-making, potentially causing HR professionals to miss crucial details or make inconsistent judgments. The challenge intensifies as the volume of job applications increases and resumes are submitted in various formats, such as Portable Document Format (PDF) and Document file format (DOCX). Additionally, the need for more standardization in resume formats makes extracting and comparing relevant information across different documents challenging. Some resumes may be poorly formatted or use unconventional layouts, further complicating the review process. Our proposed solution aims to streamline this process, allowing HR professionals to focus on more strategic tasks. The web application we have developed automates extracting and analyzing information from resumes.

II. Related Work

A Curriculum Vitae (CV) summarizes a person's educational background, employment history, and personal facts. CV forms differ greatly depending on the language, nation, and industry. They come in PDF and Word formats and usually have sections, tables, and graphic upgrades. Because of this variability, systems that match resumes have difficulties. Current systems employ optical character recognition (OCR) and natural language processing (NLP), utilizing models such as BERT for text categorization and Named Entity Recognition (NER) for necessary data extraction from intricate resumes [4]. Komanduri et al. [5] used the Tesseract OCR engine for OCR and language translation, employing adaptive binarization and contour detection for text identification. Subramanian et al. [6] combined you only look once (YOLO) for character localization with Super Resolution, achieving high word accuracy. Alamelu et al. [7] developed a machine learning and NLP system to compare resume data with job descriptions, significantly reducing resume reading time. Roy et al. [8] tackled hiring challenges with NLP and n-gram text classification, achieving 78.53% accuracy. Kinge et al. [9] used various machine learning models with 78% and 98% accuracy. Nisha B. et al. [10] created an Automated Resume Parsing and Ranking System (ARRS) using NLP for essential information extraction and candidate ranking. Amin et al. [11] introduced a semi-supervised learning system for candidate filtering, reducing manual review time. Hansen Artajaya et al. [12] designed a system to help students choose suitable internships using OCR and distance-based algorithms. Anuska M. and Umme S. M. [13] employed advanced NLP models like distilBERT and XLM for resume screening and ranking. Jayakumar et al. [14] optimized the hiring process with Random Forest, achieving 92.9% accuracy. Mohanty et al. [15] developed Resumate with the XGBoost model, achieving 96% training accuracy. Gunaseelan B. et al. [16] used machine learning techniques like XGBoost for predicting text headings and classifying resume segments. Sonali Mhatre et al. [17] achieved up to 99.48% accuracy in resume screening and ranking using convolutional neural networks (CNN) and long short-term memory (LSTM) models. Ashish Virendra Chandak et al. [18] integrated a resume parser with a job recommendation engine using machine learning and NLP for skill extraction and job matching. Muntaha Mehboob et al. [19] created a tool for candidate shortlisting and ranking, achieving 99% job-matching accuracy. Thatavarthi Giri Sougandh et al. [20] employed NLP techniques for high-accuracy resume parsing. Alderham and Jaha [21] enhanced candidate-career matching with machine learning (ML) and NLP, achieving up to 92% accuracy. Amruta Mankawade et al. [22] developed a high-accuracy job recommendation system. Senem Tanberk et al. [23] utilized OCR and NLP models, with DistilBERT and YOLOv8 enhancing recruitment efficiency.

III. Methodology

A. Overview Of the System Architecture

The proposed resume-ranking system employs a multi-tiered architecture that integrates deep learning models with traditional information retrieval techniques. This hybrid approach enables efficient resume processing and analysis, facilitating candidate-job matching. The system architecture is explained in the following sections.

1) Resume Information Extraction: HR professionals upload resumes in formats like DOCX, PDF, or images. These files are converted to JPG format for processing by the YOLOv9 model, which detects and segments text regions. Text is then extracted from these regions using EasyOCR. Subsequently, a fine-tuned mBERT model classifies the text, including personal information, education, work experience, and skills. The NER model extracts and categorizes keywords such as location and language, supplemented by regular expressions. The processed information is then stored for matching purposes.

2) Job Description Definition: Recruitment agencies or HR professionals enter the job descriptions into the system. The system lets the user set up weighting categories for different attributes used in the matching process, such as experience, education, and skills.

3) Matching and Ranking: The system employs a hybrid matching technique to align job descriptions with processed resumes, using cosine similarity to measure relevance between classified text embeddings and BM25 to match keywords such as location and language, enhancing matching accuracy. Scores are generated for each resume based on weighted categories and matching results, with resumes ranked accordingly. The ranked resumes are then presented to the user, highlighting the most relevant candidates for the job description. Fig. 1 illustrates the Resume Ranking Web Application's system components and data flow, detailing the interaction of various system modules from resume extraction to job description analysis and data storage.

B. Overview of the Dataset

The data collection process involved obtaining various resume templates from various online sources. We gathered templates from multiple royalty-free resume websites, publicly accessible professional networking platforms, and open-source resume builders. Additionally, we included templates from academic institutions' career centers that provide sample resumes for students and alumni. This approach ensured a diverse collection of 2,751 resume templates in PDF and DOCX formats, including various styles, formats, and content structures. The dataset annotation for object detection was conducted using Roboflow, a platform that provides advanced annotation tools. This paper's dataset includes only one class labeled "segment." This class denotes the various segments of text within each resume image. The dataset was divided into three sets:

- The training set comprises 75% of the dataset, totaling 3,220 images.
- The validation set comprises 19% of the dataset, totaling 819 images.
- The test set comprises 6% of the dataset, totaling 265 images.

Various preprocessing and augmentation strategies were employed through Roboflow to enhance the dataset's quality and diversity. These methods are thoroughly described in Table I.

C. Implementation

1) Object Detection: This paper explored various leading object detection models to analyze resume layouts. Facebook AI introduced DETR and Detectron2, and their designs enhance object detection capabilities. DETR leverages the transformer architecture, creating an end-to-end detection model that eliminates the need for traditional heuristics or multi-stage processes [24]. Conversely, Detectron2 is a versatile and scalable framework supporting various architectures, including Mask R-CNN and Faster R-CNN. Its adaptability and rapid development make it exceptionally practical for processing intricate images with complex layouts [25]. In YOLOv9, the

TABLE I
Image Preprocessing and Augmentation Techniques

Technique	Description
Preprocessing	
Auto-Orient	Adjust image orientation automatically
Resize	Standardize to 640 x 640 pixels
Augmentations	
Quantity	2 augmentations per image
Grayscale	Applied to 25% of images
Saturation	Adjusted between -25° and +25°
Blur	Up to 1 pixel
Noise	Up to 5.01% of pixels affected

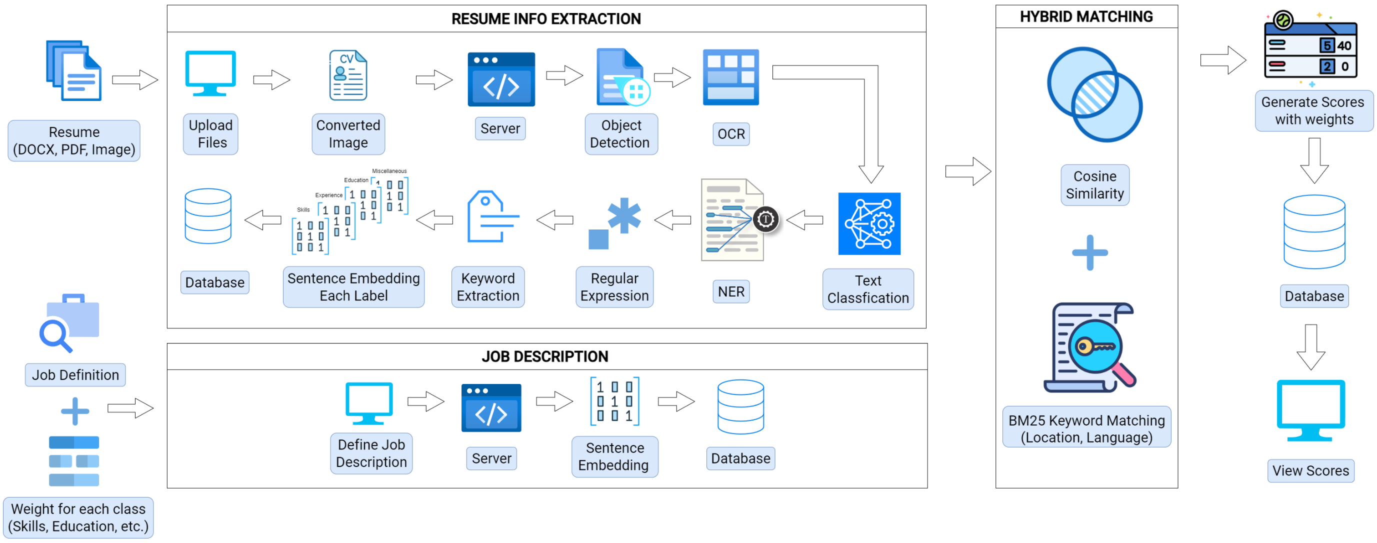

Fig. 1. Web Application Diagram

latest model in the YOLO series, innovative elements like the GELAN architecture enhance object detection. GELAN builds on the ELAN structure, allowing for varied computational blocks and enhancing flexibility and efficiency. This system combats the information bottleneck issue and enhances gradient flow using programmable gradient information (PGI), enabling state-of-the-art (SOTA) performance with reduced computational demand [26]. Based on our evaluation of different models for detecting elements in resumes, we chose YOLOv9. We compared YOLOv9, DETR, and Detectron2, trained on our custom dataset for resume layout detection. YOLOv9 stood out for its high accuracy, recall, and mean average precision (mAP) in accommodating the various resume formats. This choice aligns with recent research showing YOLO's effectiveness in handling different document layouts [23]. Being able to detect layout elements before applying OCR improves the accuracy of our resume ranking system, considering the diverse nature of resume formats. We will provide detailed results of this comparative training and evaluation process in future sections of this paper.

2) Text Recognition with OCR: We utilized EasyOCR, a trained OCR model, to extract text from the identified segments [27]. EasyOCR is renowned for its robustness and efficiency in recognizing text in various languages and fonts [28] [29]. By leveraging EasyOCR, our system can process the segmented text regions identified by the object detection model. The process begins with inputting detected text segments into the OCR model. The model then processes these segments to extract textual content, subsequently separated for the classification model.

3) Text Classification: A fine-tuned multilingual BERT (mBERT) model is utilized to classify extracted text segments. This model was sourced from the Hugging Face model repository with the usage of Transformers [30]. mBERT reduces the size of BERT's vocabulary and makes it possible to significantly reduce the model's parameters without compromising performance on downstream tasks [31]. The fine-tuned mBERT model performs better in terms of accuracy in the classification of the resume texts. The model is trained to classify text into the following 12 classes: awards, certificates, contact/name/title, education, interests, languages, para, professional experiences, projects, skills, soft skills, and summary. The first step is to provide the model with the text segments that the OCR model extracted. The model then classifies each segment into one of the predefined classes, allowing for an organized and structured representation of the resume content.

4) Zero-shot NER: We utilized GLiNER, a robust zero-shot NER model that leverages a bidirectional transformer encoder for efficient parallel entity extraction, exceeding zero-shot performance and resource efficiency across multiple NER benchmarks when compared to traditional NER models and LLMs like ChatGPT [32].GLiNER's unique capability allows for dynamic label definition during each inference, eliminating the need for retraining to accommodate new entity types. GLiNER was used to extract entities from text segments categorized by our classification model in our paper. Following classification, each text segment transferred NER to identify relevant entities, employing regular expressions for extracting specific data such as links, emails, and phone numbers. The entity labels were dynamically defined based on the classified category of each segment, as detailed in Table II.

5) Embedding Model and Hybrid Matching: We utilized the Sentence Transformers library [33] and the 'Alibaba-NLP/gte-large-en-v1.5' model from Hugging Face to generate embeddings, recognized for efficient and accurate vector embeddings for longer text sequences [34]. The system creates vector embeddings for four main categories of resume content derived from the classification model's output classes: Experience, Skills, Education, and Miscellaneous (which encompasses awards, certificates, summary, paragraphs, and in-

TABLE II
CLASSES AND CORRESPONDING NER LABELS

Class	Labels
contact/name/title	Location, person, job title
education	university, degree, date
languages	language, language level
professional experiences	company name, job title, date
summary	years of experience, designation, company name

terests).

This categorization facilitates the embedding process. Further enhancing our system's utility, users can assign weights to different resume parts—Skills, Experience, Education, Miscellaneous, and Keywords—reflecting their priorities in the job-matching process. Job descriptions are integrated into a single vector for embedding by combining details like job title, company, location, type, and required skills. The hybrid matching process is initiated based on these user-defined weights, ensuring a customized approach to aligning jobs. Let be the W_s, W_e, W_{ed}, W_m, and W_k weights for Skills, Experience, Education, Miscellaneous, and Keywords, respectively. The overall matching score S is calculated by combining cosine similarity and keyword matching scores. Motivated by the approach described in a recent publication [23], we created our hybrid matching procedure in this manner:

- Cosine Similarity Matching with Resume Parts: The cosine similarity C_i between the job description embedding and the embedding of each resume part is calculated as shown in Equation (1).

$$C_i = \frac{E_{\text{job}} \cdot E_{\text{res},i}}{\|E_{\text{job}}\| \|E_{\text{res},i}\|} \tag{1}$$

- Where E_{job} is the embedding vector of the job description, and $E_{\text{res},i}$ is the embedding vector of the i-th part of the resume. The combined cosine similarity score C is then weighted and summed in Equation (2).

$$\begin{aligned} C =& W_s \cdot C_{\text{skills}} + W_e \cdot C_{\text{experience}} + \\ & W_{ed} \cdot C_{\text{education}} + W_m \cdot C_{\text{miscellaneous}} \end{aligned} \tag{2}$$

- Keyword Matching: The BM25 algorithm [35] matches keywords, mainly focusing on location and language. Let K_{loc} and K_{lang} be the BM25 scores for location and language keywords, respectively. The keyword matching score K is calculated as in Equation (3).

$$K = 0.5 \cdot (K_{\text{loc}} + K_{\text{lang}}) \tag{3}$$

- Overall Matching Score: Finally, the overall matching score S is computed by combining the weighted cosine similarity score and the keyword matching score as Equation (4).

$$S = C + W_k \cdot K \tag{4}$$

- This method improves the system's ability to match resumes with job descriptions by looking for similar meanings and specific keywords.

6) Web Application: The web application developed for this paper emphasizes user-friendliness and detailed review with a straightforward interface that enhances usability. Users can upload up to 200 resumes, view model outputs directly, and adjust job descriptions, including defining matching weights and job parts. The application supports weight scores, job title, company location, job type, job details, and skills. It also enables detailed viewing of matching results on the interface, allowing users to review how each resume aligns with the job description comprehensively.

IV. EXPERIMANTAL RESULTS

We will begin by presenting the training results of our object detection models, which were evaluated using a custom dataset specifically created for resume layout analysis. This dataset, comprising diverse resume formats, allowed us to compare the performance of YOLOv9, DETR, and Detectron2 in accurately identifying structural elements within resumes. Following this analysis, we will detail the outcomes of implementing OCR for text extraction, along with the results of subsequent text classification and NER processes enhanced by regular expressions. Additionally, we will show an example of embedding models results in matching resumes with job descriptions, demonstrating how our multi-stage approach contributes to the resume ranking system.

A. Custom Dataset Analysis

The dataset employed in this study comprises 2,694 images, which extend to 4,304 images when data augmentations are applied. It features one class segment that is thoroughly annotated, totaling 19,111 annotations with an average of approximately 7.1 annotations per image. These annotations are systematically distributed across different subsets, with 10,898 annotations designated for training, 6,238 for validation, and 1,975 for testing. This structured distribution ensures a comprehensive training and evaluation process for the models involved. The distribution of object counts within the images is illustrated in Fig. 2.

B. Object Detection Model Training Results

The training parameters such as time, learning rate, batch size, optimizer, and other hyperparameters for each model—DETR, Detectron2, and YOLOv9—were established following the guidance from their respective research papers [24] [25] [26]. We assessed the models using metrics including Average Precision (AP) and Average Recall (AR) at varying intersections over Union (IoU) thresholds. AP measures how accurate the model is in identifying correct sections while avoiding mistakes. AR shows how well the model finds a resume's essential parts. IoU helps us understand how precisely the model outlines different resume sections. We looked at these metrics at various levels of strictness to get

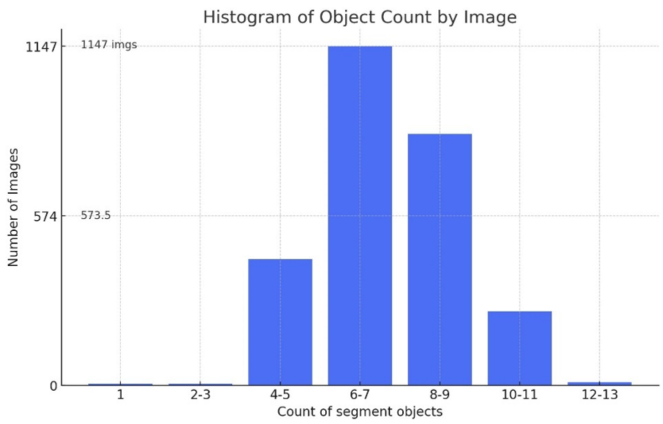

Fig. 2. Annotation Distribution

a complete picture of each model's strengths. Specifically, AP was evaluated at IoU thresholds of 0.50:0.95 (average over multiple thresholds), 0.50 (a relaxed threshold), and 0.75 (a stricter threshold). AR was assessed at IoU thresholds of 0.50:0.95 for up to 100 detections. This range of thresholds allows us to understand model performance under different levels of localization precision. The inference times on an L4 GPU using Google Colab showed DETR at 1.05 seconds, Detectron2 at 1.17 seconds, and YOLOv9 at a notably faster 0.24 seconds. This evaluation highlighted YOLOv9's best precision at higher IoU thresholds and quicker processing, making it ideal for precise and real-time resume segmentation tasks. The performance of each model was assessed using the metrics mentioned above. The results are summarized in Table III.

C. *Text Processing and Analysis Results*

1) OCR Results: While EasyOCR generally demonstrated high accuracy in text recognition across various sections of resumes, our evaluation identified specific issues that could affect the quality of data extraction and, consequently, the accuracy of subsequent text processing stages. These issues included character misinterpretations, such as confusing 'O' with '0', 'l' with '1', or 'S' with '5'. In some cases, we observed more complex misreads, like interpreting "Manager" as "Mana9er" or "Experience" as "Exper1ence". Such errors can significantly alter the meaning of important resume details, potentially leading to misclassifying job titles or skills. Additionally, punctuation errors, where semicolons were mistakenly used instead of commas or periods misinterpreted as commas, could change the meaning of the data or disrupt the syntax, affecting further parsing. Spacing issues, like unnecessary spaces within words—for example, "high-quality" appearing as "high - quality"—or the merging of separate words like "softwareengineering", might impact the text classification and entity recognition processes, potentially leading to inaccuracies in the NER process. Font variations, especially in stylized resume headers or logos, occasionally resulted in complete misinterpretation of text, such as the company name "TechInnovate" being read as "TechInn0v8". These findings highlight the need for implementing post-processing correction mechanisms or conducting manual reviews in some cases to ensure data integrity before it progresses to the classification and NER stages. The frequency and type of OCR errors varied across different resume formats and designs, suggesting that a diverse dataset of resume layouts is crucial for developing robust correction mechanisms.

2) Text Classification Results: The text classification stage follows the extraction of text via the OCR process. For this purpose, we utilized a pre-trained model from Hugging Face [36], specifically designed to classify text sections within resumes. The model was evaluated on an unknown resume dataset to assess its generalizability and performance. The performance metrics for the model are detailed in Table IV. Using a pre-trained model offers the advantage of leveraging existing knowledge learned from a large corpus of resume data, potentially improving classification accuracy across diverse resume formats. However, it is important to note that the model's performance may vary depending on how closely the unknown dataset aligns with the distribution of resumes used in the original training set.

These metrics indicate high accuracy and reliability in the model's ability to classify text into predefined categories such as certificates, contact information, education, languages, professional experiences, and skills. Table V illustrates the model's classification output using text extracted from resumes. These results highlight the model's accuracy in categorizing text. The model classifies text from various resume sections, including standard and less common categories. This classification step prepares the data for the subsequent named entity recognition process.

3) NER Results: Following the text classification stage, the next phase involves extracting specific entities from the classified text using the GLiNER model. The performance of the NER model is not isolated from the preceding stages of data processing, notably the output from the OCR. Table VI

TABLE III
OBJECT DETECTION MODELS EVALUATION RESULTS

Metric	DETR	Detectron2	YOLOv9
AP[IoU=0.50:0.95] (all)	0.730	0.731	**0.850**
AP[IoU=0.50] (all)	0.925	0.930	**0.965**
AP[IoU=0.75] (all)	**0.833**	0.824	-
AR[IoU=0.50:0.95] (all)	**0.790**	0.783	-
Inference Time (per image)	1.05s	1.17s	**0.24s**

TABLE IV
CLASSIFICATION MODEL PERFORMANCE METRICS [36]

Metric	Value
Loss	0.0369
F1 Score	0.9652
ROC AUC	0.9808
Accuracy	0.9621

TABLE V

TEXT CLASSIFICATION RESULTS

Classified Class	Extracted Texts
Certificates	COURSES Generative Design for Industrial App. Autodesk (Coursera) 2019
contact/name/title	Dhriti Advani Design Engineer PERSONAL INFO Address New Delhi; India Phone 9958906580 Email advanidzz@gmail.com
education	EDUCATION B.Tech Design Eng. |2017 BITS Pilani
languages	LANGUAGES Hindi English
professional_experiences	EMPLOYMENT HISTORY Industrial Designer Product Design Aug 2019...
skills	SKILLS 3D CAD Software Product Packaging Design Adobe Illustrator...
summary	Design engineering professional adept at documenting products...

illustrates the model's classification output using text extracted from resumes. Our analysis shows that the GLiNER model produces average good results in entity recognition but consistently yields lower confidence scores. Importantly, it should be noted that this NER model is implemented using a zero-shot approach, meaning it has not been specifically trained on resume data. This zero-shot nature likely contributes to the observed lower confidence scores, as the model attempts to identify entities in a domain for which it was not explicitly trained. Despite this challenge, the model's ability to produce averagely good results demonstrates the potential of zero-shot learning in specialized contexts like resume parsing. The lower confidence scores may also be attributed to other factors, such as the complexity and variability of resume formats or potential OCR errors from previous stages.

4) Matching Results: Selecting a suitable embedding model is crucial for accurate and efficient resume-job matching. We utilized the MTEB Benchmark, which evaluates models based on performance with longer sequences, average evaluation scores, and model size. After review, we selected the gte-large-en-v1.5 model [34] [37]. This model was chosen for its superior handling of lengthy texts and optimal balance between performance and size [38] [39] [40], making it ideal for our application. This model excels in processing complex job descriptions and resume data, ensuring accurate embeddings for similarity and relevance comparisons [41] [42] [43]. Below, we present the match results, including scores for semantic attributes and BM25 keyword matching [44] [45]. Table VII demonstrates the flexible weighting system, showing how candidate qualifications align with job requirements based on adjustable attribute importance.

V. CONCLUSION AND FUTURE WORK

In this study, we developed an innovative resume ranking web application that leverages deep learning techniques to improve recruitment processes. Our system integrates state-of-the-art machine learning models, including YOLOv9 for

TABLE VI

NER AND REGULAR EXPRESSIONS RESULTS

Class	Text	Label	Score	Method
cont/name/title	Dhriti Advani	Person	0.9989	NER
	Design Engineer	Job Title	0.8723	NER
	New Delhi; India	Location	0.5126	NER
	9958906580	Phone Number	-	regex
Education	B.Tech Design Engineering	University	0.3845	NER
Languages	Hindi	Lang.	0.9688	NER
	English	Lang.	0.9305	NER
Prof. Exp.	Industrial Designer	Job Title	0.6581	NER
	Aug 2019	Date	0.5382	NER
	BOSE India, Gurgaon	Company Name	0.3674	NER
	Industrial Desig. Eng.	Job Title	0.9583	NER
	Jan 2017	Date	0.5085	NER
	Amazon India_ New Delhi	Company Name	0.3086	NER
Summary	Design eng. prof.	Desig.	0.6279	NER
	design eng.	Desig.	0.3344	NER

TABLE VII

HYBRID MATCHING SCORES FOR AN EXAMPLE SCENARIO

Resume ID	Skills (0.2)	Experience (0.4)	Education (0.1)	Misc. (0.2)	Keyword (0.1)	Final Score
4	0.14	0.296	0.07	0.146	0	**0.65**
5	0.12	0.298	0.047	0.146	0	**0.61**
3	0.118	0.277	0.0	0.141	0.05	**0.59**
6	0.148	0.273	0.062	0.108	0	**0.59**
7	0.146	0.273	0.0	0.144	0	**0.56**
1	0.1	0.2	0.03	0.108	0.05	**0.44**
2	0.103	0.209	0.05	0.078	0	**0.44**
8	0.121	0.275	0.048	0.0	0	**0.44**

object detection, EasyOCR for text recognition, fine-tuned mBERT for text classification, and GLiNER for named entity recognition, enabling extraction, categorization, and matching of resume information with job descriptions. Key achievements include implementing a multi-model approach for comprehensive resume parsing, utilizing cosine similarity and BM25 algorithms for precise matching, and developing an intuitive interface with adjustable weighting categories for HR professionals. Despite these successes, we encountered challenges such as OCR accuracy issues with varying fonts and layouts, occasional inaccuracies in GLiNER's zero-shot NER approach, and limitations in capturing whole semantic meaning with dense vector similarity. We propose several future directions to address these challenges and further enhance our system. These include exploring advanced OCR models and post-processing techniques, creating a specialized NER dataset for resumes, refining information extraction methods, incorporating hybrid vector techniques for improved semantic matching, and expanding functionality to benefit job seekers.

We also aim to update our resume dataset regularly, implement data augmentation techniques, and optimize model performance for real-time processing. We aim to create a better tool for the recruitment industry by making these improvements. This tool will be more accurate, adaptable, and efficient, benefitting HR professionals and job seekers by improving the hiring process.

REFERENCES

[1] Baykal, Elif. (2020). Digital Era and New Methods for Employee Recruitment. 10.4018/978-1-7998-1125-1.ch018.

[2] S. Solanki and Dr. Gujarati, "The Digital Revolution In Recruitment: Unraveling The Impact And Challenges Of E-Recruitment," Educational Administration Theory and Practices, vol. 30, Jun. 2024, doi: 10.53555/kuey.v30i6(S).5362.

[3] Rozario, S. D., Venkatraman, S., & Abbas, A. (2019). Challenges in recruitment and selection process: An empirical study. Challenges, 10(2), 35.

[4] Palshikar, G. K., Pawar, S., Banerjee, A. S., Srivastava, R., Ramrakhiyani, N., Patil, S., ... & Chalavadi, D. (2023). RINX: A system for information and knowledge extraction from resumes. Data & Knowledge Engineering, 147, 102202.

[5] Komanduri, S., Roopa, Y. M., & Bala, M. M. (2019, March). Novel approach for image text recognition and translation. In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) (pp. 596-599). IEEE.

[6] Subramanian, S., Kekatpure, V., Raymond, G., Parab, K., Dugad, S., & Shirke, A. (2022, January). TEYSuR-Text Extraction with YOLO and Super Resolution. In 2022 International Conference for Advancement in Technology (ICONAT) (pp. 1-7). IEEE.

[7] M. Alamelu, D. S. Kumar, R. Sanjana, J. S. Sree, A. S. Devi and D. Kavitha, "Resume Validation and Filtration using Natural Language Processing," 2021 10th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON), Jaipur, India, 2021, pp. 1-5, doi: 10.1109/IEMECON53809.2021.9689075

[8] Roy, P. K., Chowdhary, S. S., & Bhatia, R. (2020). A Machine Learning approach for automation of Resume Recommendation system. Procedia Computer Science, 167, 2318-2327.

[9] Kinge, B., Mandhare, S., Chavan, P., & Chaware, S. M. (2022). Resume Screening using Machine Learning and NLP: A proposed system. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 8(2), 253-258.

[10] Nisha, B., Manobharathi, V., Jeyarajanandhini, B., & Sivakamasundari, G. (2023, December). HR Tech Analyst: Automated Resume Parsing and Ranking System through Natural Language Processing. In 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS) (pp. 1681-1686). IEEE.

[11] Amin, S., Jayakar, N., Sunny, S., Babu, P., Kiruthika, M., & Gurjar, A. (2019, January). Web application for screening resume. In 2019 International Conference on Nascent Technologies in Engineering (ICNTE) (pp. 1-7). IEEE.

[12] Artajaya, H., Giancarlos, J., Moniaga, J. V., & Chowanda, A. (2024, February). Job Recommendation System based on Resume using Natural Language Processing and Distance-based Algorithm. In 2024 IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS) (pp. 1-6). IEEE.

[13] Mukherjee, A. (2024, February). Resume Ranking and Shortlisting with DistilBERT and XLM. In 2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE) (pp. 301-304). IEEE.

[14] Jayakumar, N., Maheshwaran, A. K., Arvind, P. S., & Vijayaragavan, G. (2023, April). On-Demand Job-Based Recruitment For Organisations Using Artificial Intelligence. In 2023 International Conference on Networking and Communications (ICNWC) (pp. 1-6). IEEE.

[15] Mohanty, S., Behera, A., Mishra, S., Alkhayyat, A., Gupta, D., & Sharma, V. (2023, May). Resumate: A prototype to enhance recruitment process with NLP based resume parsing. In 2023 4th International Conference on Intelligent Engineering and Management (ICIEM) (pp. 1-6). IEEE.

[16] Gunaseelan, B., Mandal, S., & Rajagopalan, V. (2020, December). Automatic extraction of segments from resumes using machine learning. In 2020 IEEE 17th India Council International Conference (INDICON) (pp. 1-6). IEEE.

[17] Mhatre, S., Dakhare, B., Ankolekar, V., Chogale, N., Navghane, R., & Gotarne, P. (2023, June). Resume Screening and Ranking using Convolutional Neural Network. In 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS) (pp. 412-419). IEEE.

[18] Chandak, A. V., Pandey, H., Rushiya, G., & Sharma, H. (2024, February). Resume Parser and Job Recommendation System using Machine Learning. In 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC) (pp. 157-162). IEEE.

[19] Mehboob, M., Ali, M. S., Islam, S. U., & Ali, S. S. (2022, October). Evaluating automatic CV shortlisting tool for job recruitment based on machine learning techniques. In 2022 Mohammad Ali Jinnah University International Conference on Computing (MAJICC) (pp. 1-4). IEEE.

[20] Sougandh, T. G., Reddy, N. S., & Belwal, M. (2023, November). Automated Resume Parsing: A Natural Language Processing Approach. In 2023 7th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS) (pp. 1-6). IEEE.

[21] Alderham, A. H., & Jaha, E. S. (2022, December). Comparative Semantic Resume Analysis for Improving Candidate-Career Matching. In 2022 14th International Conference on Computational Intelligence and Communication Networks (CICN) (pp. 313-321). IEEE.

[22] Mankawade, A., Pungliya, V., Bhonsle, R., Pate, S., Purohit, A., & Raut, A. (2023, April). Resume Analysis and Job Recommendation. In 2023 IEEE 8th International Conference for Convergence in Technology (I2CT) (pp. 1-5). IEEE.

[23] Tanberk, S., Helli, S. S., Kesim, E., & Cavsak, S. N. (2023, September). Resume Matching Framework via Ranking and Sorting Using NLP and Deep Learning. In 2023 8th International Conference on Computer Science and Engineering (UBMK) (pp. 453-458). IEEE.

[24] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, August). End-to-end object detection with transformers. In European conference on computer vision (pp. 213-229). Cham: Springer International Publishing.

[25] Kirillov, A., Wu, Y., He, K., & Girshick, R. (2020). Pointrend: Image segmentation as rendering. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9799-9808).

[26] Wang, C. Y., Yeh, I. H., & Liao, H. Y. M. (2024). YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. arXiv preprint arXiv:2402.13616.

[27] JaidedAI. (2023). EasyOCR. GitHub. Retrieved from https://github.com/JaidedAI/EasyOCR

[28] Vedhaviyassh, D. R., Sudhan, R., Saranya, G., Safa, M., & Arun, D. (2022, December). Comparative analysis of easyocr and tesseractocr for automatic license plate recognition using a deep learning algorithm. In 2022 6th International Conference on Electronics, Communication and Aerospace Technology (pp. 966-971). IEEE.

[29] Wu, X., Luo, C., Zhang, Q., Zhou, J., Yang, H., & Li, Y. (2019). Text Detection and Recognition for Natural Scene Images Using Deep Convolutional Neural Networks. Computers, Materials & Continua, 61(1).

[30] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2019). Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

[31] Abdaoui, A., Pradel, C., & Sigel, G. (2020). Load what you need: Smaller versions of multilingual BERT. In Proceedings of SustaiNLP / EMNLP.

[32] Zaratiana, U., Tomeh, N., Holat, P., & Charnois, T. (2023). Gliner: Generalist model for named entity recognition using a bidirectional transformer. arXiv preprint arXiv:2311.08526.

[33] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. http://arxiv.org/abs/1908.10084

[34] Li, Z., Zhang, X., Zhang, Y., Long, D., Xie, P., & Zhang, M. (2023). Towards general text embeddings with multi-stage contrastive learning. arXiv preprint arXiv:2308.03281.

[35] Brown, D., Jain, S., Novotný, V., & nlp4whp. (2022). rank_bm25. Zenodo. https://doi.org/10.5281/zenodo.4520056

[36] Abida, H. (2022). distilBERT-finetuned-resumes-sections. Hugging Face. Retrieved from https://huggingface.co/has-abi/distilBERT-finetuned-resumes-sections

[37] Dwyer, B., Nelson, J., Hansen, T., et al. (2024). Roboflow (Version 1.0) [Software]. Available from https://roboflow.com
[38] Muennighoff, N., Tazi, N., Magne, L., & Reimers, N. (2022). MTEB: Massive text embedding benchmark. arXiv preprint arXiv:2210.07316.
[39] Wadi, m., Shobole, A., Elmasry, W., & Kucuk, I. (2024). Load frequency control in smart grids: A review of recent developments. Renewable and Sustainable Energy Reviews, Volume 189, Part A. https://doi.org/10.1016/j.rser.2023.114013.
[40] Wadi, M., Elmasry, W., Shobole, A., Tur, M. R., Bayindir, R., & Shahinzadeh, H. (2021). Wind Energy Potential Approximation with Various Metaheuristic Optimization Techniques Deployment. 2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS), Tehran, Iran, Islamic Republic of, (pp. 1-6). doi: 10.1109/ICSPIS54653.2021.9729389.
[41] Sahmoud, S., Abudalfa, S., & Elmasry, W. (2022). AT-ODTSA: a Dataset of Arabic Tweets for Open Domain Targeted Sentiment Analysis. International Journal of Computing and Digital Systems, 11(1), 1299-1307.
[42] Wadi, M., Elmasry, W., & Tamyigit, F. A. (2023). Important considerations while evaluating wind energy potential. Journal of the Faculty of Engineering and Architecture of Gazi University, 38(2), 947-962.
[43] Elmasry, W., Wadi, M., & Shahinzadeh, H. (2022, May). Two-tier cascaded classifiers to improve electrical power quality. In 2022 26th International Electrical Power Distribution Conference (EPDC) (pp. 96-101). IEEE.
[44] Wadi, M., & Elmasry, W. (2023). A comparative assessment of five different distributions based on five different optimization methods for modeling wind speed distribution. Gazi University Journal of Science, 36(3), 1096-1120.
[45] Wadi, M., Elmasry, W., Kucuk, I., & Shahinzadeh, H. (2022, November). Sensitivity reliability analysis of power distribution networks using fuzzy logic. In 2022 12th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 190-195). IEEE.