An Approach Based on Named Entity Recognition and Semantic Analysis for Recruitment Efficiency and Optimization

Ismail Ifakir[1], Noureddine Mohtaram[2], El Habib Nfaoui[3], Abderrahim Zannou[4], Mohammed El Hassouni[5]
L3IA Laboratory, Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco[1,3]
LRIT Rabat IT center, Faculty of Sciences, Mohammed V University in Rabat, Morocco[2]
ERC2IA, Faculty of Sciences and Techniques, Al Hoceima, Abdelmalek Essaadi University, Tetouan, Morocco[4]
FLSH, Mohammed V University in Rabat, Morocco[5]

Abstract*—Modern recruitment requires smarter, faster, and more inclusive methods to manage the growing volume and diversity of job applications and candidate resumes. Manual screening is often ineffective and unreliable especially in low-resource or multilingual contexts. To address this challenge, we propose an approach that automates and optimizes key stages of the recruitment process. This three-stage approach includes: 1) extracting structured data from resumes using a robust Named Entity Recognition (NER) system, which comprises a NER annotator, a feature extractor, and a transition-based parser; 2) employing a fine-tuned transformer model to perform semantic matching between candidates and job descriptions; and 3) leveraging a large language model to generate interview questions tailored to specific job requirements, thereby improving the relevance and personalization of candidate assessments. The recruitment system was tested on a large-scale resume and job posting dataset across multiple domains. Our NER model reported an F1-score of 85.11% in entity extraction, and the matching component reported accuracy levels as high as 92% when using hierarchical job classes. The results prove the efficacy of combining deep learning techniques with semantic reasoning in enhancing automation, accuracy, and fairness in hiring.

Keywords*—*Named entity recognition; large language models; feature extraction; generate question; matching

I. Introduction

With technology advancing rapidly and professional fields becoming increasingly diverse, companies and recruitment systems are now confronted with an unprecedented volume of job applications. Large firms and headhunters process hundreds of Curriculum Vitae (CVs) each day, with each document exhibiting unique formatting styles, layouts, and structures. This variability poses a significant challenge for automated recruitment systems, which must efficiently extract, organize, and evaluate candidate information at scale. Named Entity Recognition (NER), a subtask of Natural Language Processing (NLP), offers one of the most promising solutions to this challenge. NER [1], [2], [3], [4], [5], [6], [7], [8], [9] refers to the process of identifying and classifying named entities in text into predefined categories including names, organizations, locations, dates, and other specific terms. NER has shown notable success in extracting structured knowledge from unstructured resume content [10], [11], [12], [13], [14], [15], in the framework of resume analysis. It helps important components name, contact information, educational background, professional experience, technical and soft skills, and certifications to automatically detect and organize. Another crucial component of intelligent recruitment is the task of matching candidates with job descriptions [16], [17], [18], [19], [20], [21], which aims to automatically align candidate profiles with employer requirements. The objective is to assess how well a candidate's qualifications, background, and skills correspond to the expectations and preferences outlined in job postings. Automating this matching process can significantly enhance the quality of candidate selection, reduce time-to-hire, and improve overall recruitment efficiency. However, this remains a complex and inherently challenging task. It goes beyond simply identifying similar keywords or job titles; it requires an understanding of the semantic context in both resumes and job descriptions. Terminologies often differ, experiences may only partially align, and important qualifications may be implied rather than explicitly stated. As a result, traditional keyword-based methods are frequently inadequate. To achieve accurate and meaningful matching, advanced techniques such as language embeddings, semantic similarity modeling, and context-aware entity alignment are increasingly being employed. Following the candidate–job matching phase, companies often need to conduct personalized evaluations of shortlisted applicants. At this stage, recruiters must delve deeper into specific aspects of each profile, which typically involves asking targeted follow-up questions. However, many organizations lack the time and resources to manually craft relevant questions for every candidate. To address this, the automatic generation of entity-driven interview questions has emerged as a valuable solution [22]. By leveraging entities extracted from resumes and job descriptions during the matching phase, the system can generate tailored questions for each candidate, helping recruiters verify key information or probe specific skills and experiences. For example, based on the entity work experience, the system might generate questions such as: Can you elaborate on your responsibilities at Company X? or What challenges did you face during your role as a Data Analyst? Thus enabling a more focused, consistent, and efficient evaluation process.

To address these challenges, we propose an approach that automates and optimizes key stages of the recruitment process. The process begins with the extraction of structured information from resumes using a robust Named Entity Recognition framework, composed of an annotator, a feature extraction, and a transition-based parser. This is followed by the use of a fine-tuned transformer-based model to achieve semantic alignment between candidate profiles and job descriptions. Finally, leveraging a large language model to generate interview questions tailored to specific job requirements. This recruitment process mitigates the limitations of traditional recruitment workflows by automating resumes parsing, CVs-job matching, and question generation ultimately reducing manual effort, improving selection accuracy, and enhancing the overall efficiency of the recruitment process for both employers and applicants.

This study is structured as follows generally: Section II discusses related work, stressing developments in entity extraction from CVs and other matching techniques applied in recruitment. Section III covers the preliminaries and details on the models applied in this work. The suggested framework is presented in Section IV together with information on extracting important entities, matching between candidates and job descriptions, and entity-based interview questions. Section V addresses the experimental results, evaluating the performance of the framework in relation to current models and so analyzing its possible influence and performance. Section VI ends the work by aggregating important contributions and suggests future avenues of research to improve recruiting automation.

II. Related Work

In the field of CV parsing, many companies process large volumes of resumes to identify suitable candidates for employment, making manual analysis increasingly impractical. As a result, significant progress has been made in Named Entity Recognition for extracting structured information from resumes [23], [24], as well as in developing algorithms that efficiently match candidates with job requirements. Several approaches have been proposed to tackle these problems. For instance, Pham et al. [25] proposed a deep learning-based NER model for resume parsing in their paper “Study of Information Extraction in Resume”. Their method involves four main steps: text normalization, rule-based NER, which based on deep neural networks, and text segmentation. Using a 1,000 manually labeled Vietnamese resume medium-sized dataset, they extracted entities such as names, addresses, talents, degrees, experience, and universities. Zhu et al. [26] proposed a user behavior-based preference alignment framework for fine-tuning LLMs to improve job recommendations through resume completion. To address noise in behavioral data (i.e., bias and variance), they introduced a noise-robust LLM alignment method named Denoised Direct Preference Optimization (Denoised DPO), which disentangles genuine user preferences from noisy data. Specifically, they designed a novel reward function for preference estimation by combining an LLM-based component for real user preference with a regression model for bias disentanglement. In addition, they developed a Thurstonian-style model for job-seekers’ preference modeling to stabilize data reliability amidst behavior variances. Extensive offline and online experiments demonstrated the effectiveness of this approach in enhancing recommendation quality.

Vanetik and Kogan [27] improved resume-job matching using semantic similarity by taking the assistance of BERT-based sentence embeddings. They extracted entities like job titles, credentials, and skills using SpaCy’s NER component, and employed keyword extraction using TF-IDF. The integration of keyword-based methods, BERT embeddings, and NER formed a solid base for automatic candidate ranking. Bakliwal et al.[28] addressed the problem of orphan entity resolution using a knowledge graph-based approach and models such as BERT and RoBERTa. Their model, via concept mining, association mining, and NER, detected and contextualized doubtful resume entries. The inferred knowledge graphs linked these entities to external knowledge bases, improving contextual comprehension and candidate-job matching. Wang et al. [29] examined the impact of pretrained language models, including BERT, ERNIE, ERNIE2.0-tiny, and RoBERTa, on NER performance. By fine-tuning all the models, they evaluated the effect of different pretraining strategies on extraction of entities from unstructured text and offered valuable information on model selection for NER applications. Tran et al. [30] proposed a practical solution to occupational skill identification in Vietnamese job advertisements in a bid to address the skill mismatch most prevalent in the domestic labor market. Rather than approach skill detection as a standard NER task, they approached it as ranking. Identified phrases were ranked by semantic similarity with context neighbors without involving extensive annotated datasets or supervised NER models. Rosenberger et al. [31] introduced CareerBERT, a deep learning-based model for resume-to-job matching enhancement. The model acquires the ability to update and adapt to dynamic labor market shifts using unstructured resume and job posting data. CareerBERT integrates ESCO taxonomy and EURES job posting data into an active and structured job corpus. The accuracy and validity of recommended jobs are improved as a consequence. Tyagi et al. [32] countered gendered bias in resume matching systems using word embedding refinement and improving classification algorithms. Their system demonstrates how gendered words, descriptions, and skill mentions can produce biased representations in resumes that result in unequal job classification. Using real-world datasets, they demonstrated high correlation between embedding bias and occupational gender skewing and proposed debiasing techniques to decrease the bias. Lastly, Barducci et al. [33] addressed the problem of resume information extraction in the Italian labor market by proposing an effective end-to-end framework. Their system provides a complete candidate overview, including personal information, skills, and work experiences. The framework extracts raw data from resumes and segments them into semantically consistent parts using linguistic patterns. Each segment is then processed with a NER algorithm, based on pre-trained language models, to extract the most relevant information.

Previous studies have demonstrated a wide range of approaches aimed at addressing entity extraction challenges in resume parsing and job matching. While substantial progress has been made such as generating evaluation questions from resumes, identifying resumes that closely match specific job descriptions, and aligning job postings with suitable candidates significant gaps remain. These include the underutilization of advanced deep learning models, limited integration of semantic embeddings, and insufficient application of knowledge graphs for contextual understanding. To address these limitations, this study proposes a more comprehensive approach to entity extraction from resumes using a NER-based framework. The goal is to enable more accurate and context-aware parsing of candidate profiles, facilitate scalable and effective matching between resumes and job descriptions, and automatically generate relevant interview questions based on extracted entities.

III. Preliminaries

This section provides the necessary background to support the rest of the study, including named entity recognition, text embeddings, pre-trained transformer-based models from the BERT family, and large language models.

A. Named Entity Recognition

Named Entity Recognition [1] is a core task in natural language processing that involves identifying and classifying entities in text into predefined categories such as persons, organizations, locations, dates, and others. NER systems typically rely on sequence labeling techniques, where each token is assigned a tag indicating whether it belongs to an entity and what type it represents. NER is widely applied in various real-world scenarios, including information extraction from unstructured text, question answering, content classification, knowledge graph construction, and automated resume analysis. In the recruitment domain, for example, NER is essential for extracting structured information from resumes and job descriptions, such as candidate names, educational qualifications, professional experience, and specific skills. Modern NER approaches leverage contextual embeddings and transformer-based architectures to achieve high accuracy and robustness, particularly in domain-specific or noisy text environments.

B. Embeddings

Embeddings are numerical models of unprocessed data text or category values, for example into continuous vector spaces. Basic to deep learning models, reflecting semantic linkages and similarities between data points unlike traditional discrete or one-hot representations is not a secret. Mostly for the preservation of pertinent information, embeddings convert high-dimensional data into dense, lower-dimensional representations. This ensures more successful and efficient application for jobs involving classification, grouping, and similarity analysis. By grouping semantically or structurally similar objects closer together in the embedding space, embeddings let machine learning algorithms find underlying patterns in the data and overcome expressly specified constraints.

C. BERT Family

Processing text bidirectionally, the deep learning model BERT (Bidirectional Encoder Representations from Transformers) [34] considers both left and right contexts concurrently. It discovers contextual links between words inside a sentence by means of Transformer architecture and more importantly, self-attention methods. For many NLP uses, large corpus pre-training helps BERT provide highly contextualized embeddings. In fields including Named Entity Recognition, it has greatly increased performance, sometimes exceeding historical benchmarks. By means of fine-tuning, BERT can be customized to domain-specific tasks even with limited labeled data.

Under the Masked Language Modeling aim, Bert is trained whereby the model is taught to forecast random words in a phrase by masking them. Moreover, next sentence prediction (NSP) improves sentence-level comprehension. Though more contemporary models like RoBERTa eliminate the NSP aim, BERT's multi-layer Transformer encoders capture deep language patterns, making it quite well-suited for NER, information extraction, and automated text processing applications. Dynamic masking eliminates the NSP aim and produces a superior variant of BERT, RoBERTa, thereby, enhancing the pretraining process. Unlike BERT's fixed masking, RoBERTa exposes the model to a wider diversity of linguistic settings, as masking patterns vary on the same input across training epochs. From this, in NER, resulting more solid representations and better task performance. RoBERTa has shown more accuracy and efficiency in named entity extraction, however its architecture is essentially similar BERT.

By means of disentangled attention via DeBERTa (Decoding-enhanced BERT with disentangled attention), one obtains important benefits by separating content and spatial embeddings. While traditional models mix lexical meaning and spatial information, DeBERTa especially separates these elements thereby improving the model's grasp of word order and sentence structure. This method shows especially good performance in entity recognition tasks, where exact capture of subtle dependencies is essential. DeBERTa also features a new decoding technique designed to raise token prediction accuracy.

More recently, CamemBERT a multilingual BERT-based model tailored for the labor market domain, has shown strong performance in skill and occupation extraction tasks. It is trained using job-related taxonomies and multilingual job corpora to enhance cross-lingual generalization and semantic understanding in employment-related text. By specializing in the semantics of job postings and CVs, CamemBERT proves particularly effective for both skill detection and entity recognition in multilingual or non-standard CV formats.

D. Large Language Models

LLMs are advanced neural network architectures trained on massive text corpora to understand and generate human-like language. Built primarily on transformer architectures, LLMs utilize self-attention mechanisms to model complex linguistic patterns, enabling them to perform a wide range of natural language processing tasks such as summarization, translation, question answering, and dialogue generation. Their ability to generate contextually relevant and coherent responses makes them suitable for integration into applications requiring intelligent language understanding and generation. Notable examples of LLMs include Google Gemini developed by Google DeepMind for advanced language understanding and generation, and Meta's LLaMA (Large Language Model Meta AI) designed as an efficient, open-weight model for research and practical applications. These models represent key milestones in the evolution of generative AI and have significantly contributed to progress across a variety of domains.

IV. Proposed Approach

This section describes the method adopted to enhance and automate the recruitment process. The approach leverages NER for entity extraction, candidate–job matching, and entity-based question generation.

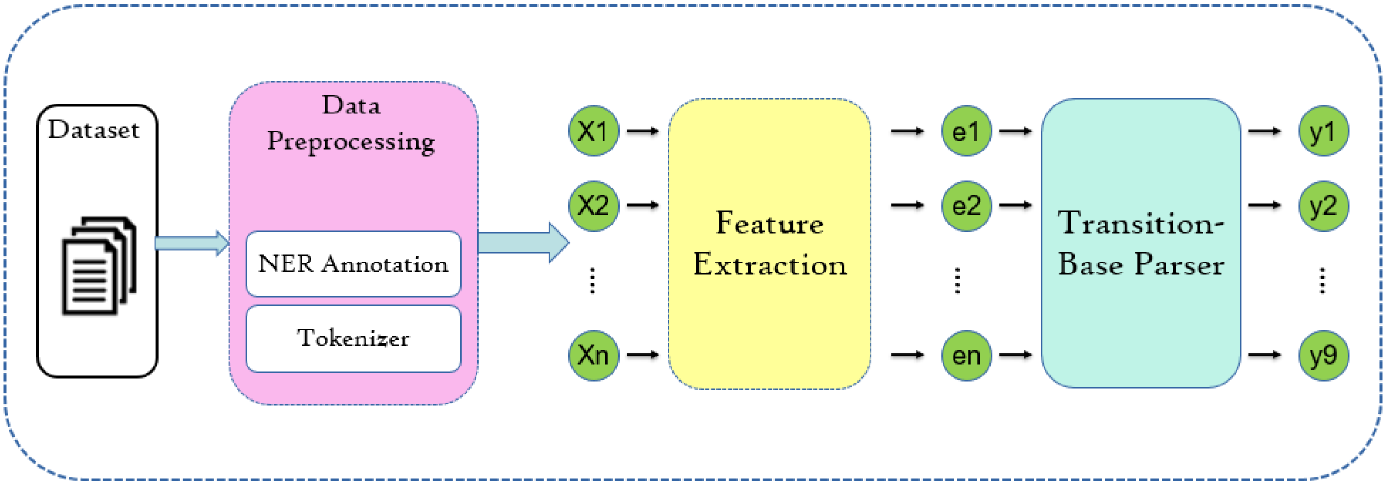

Fig. 1. Overview of the proposed NER pipeline architecture.

A. NER-Based Entity Extraction

The structure of the NER entity extraction model is illustrated in Fig. 1. This framework is composed of five key modules: data collection, preprocessing, feature extraction, and the transition-based parser. To effectively leverage these components for feature learning and entity recognition, the process begins with model representation. This involves collecting a dataset of resumes in various formats and converting them into machine-readable text. The preprocessing stage consists of two essential tasks: Named Entity Recognition annotation and tokenization. Following that, a pre-trained transformer model processes the tokens to generate contextual embeddings that capture semantic relationships between words. Finally, the transition-based parser applies a sequence of learned transitions to progressively predict entity boundaries and assign labels, enabling the extraction of structured entities from unstructured resume text.

1) Data preprocessing: From multiple sources including a broad spectrum of companies, job vacancies, and skill levels we assembled a diversified collection of resumes in PDF and Word formats. These resumes are written in English and follow a consistent structure, enabling uniform data representation. We denote the dataset as $D = \{d_1, d_2, \ldots, d_n\}$, where D represents the collection of n resumes, and each d_i is a raw document. For more details about the dataset, refer to Section V.

Each document d_i in the dataset is first manually or semi-automatically annotated with named entities directly on the raw text. These annotations are created as labeled character spans and stored as triplets: $A_i = \{(s_k, e_k, l_k)\}$. Here, s_k and e_k denote the character-level start and end offsets of an entity, and l_k is the corresponding entity label (e.g., `AGE`, `SKILLS`).

After annotation, the raw text is passed through spaCy's rule-based tokenizer, which segments it into a sequence of tokens: $T_i = \{t_{i1}, t_{i2}, \ldots, t_{im}\}$. During training, the character-based entity spans A_i are automatically aligned with the token boundaries defined in T_i, producing token-level supervision for the NER model.

2) Contextual embedding: Once tokenized, the sequence T_i is passed into a pre-trained transformer model such as RoBERTa, via the spaCy-transformers integration. This model generates a contextual embedding for each token:

$$E_i = \text{Em}(T_i) = \{\mathbf{e}_{i1}, \mathbf{e}_{i2}, \ldots, \mathbf{e}_{im}\}, \quad \mathbf{e}_{ij} \in \mathbb{R}^d \tag{1}$$

These contextual embeddings in Eq. (1) encode each token with rich semantic and syntactic information by leveraging sentence-level dependencies and multi-head attention mechanisms.

3) Transition-based parser: The final step is handled by the Transition-Based Parser module, which consumes the token vectors and predicts entity boundaries and labels through a sequence of learned transitions. At each state s, the model scores and selects the most probable action a_s, as shown in Eq. (2):

$$a_s = \arg\max_{a \in A} f_{\text{score}}(\sigma_s, a) \tag{2}$$

where, σ_s is the parser state, A is the set of possible transition actions (e.g., BEGIN, CONTINUE, OUT), and f_{score} is the internal scoring function (a feed-forward network with maxout layers). The final labeled spans are stored in `doc.ents` for downstream use.

B. Matching

Fig. 2 illustrates the overall architecture of the matching process between candidates and job descriptions. It consists of four main components: Input Data, Entity Extraction, Feature Extraction, and Similarity Matching. The process begins with unstructured input data, namely job descriptions and candidate resumes written in French.

In the first step, entities are extracted using a custom NER model, as detailed in Fig. 1. This model identifies key elements such as skills, technologies, and job-specific attributes

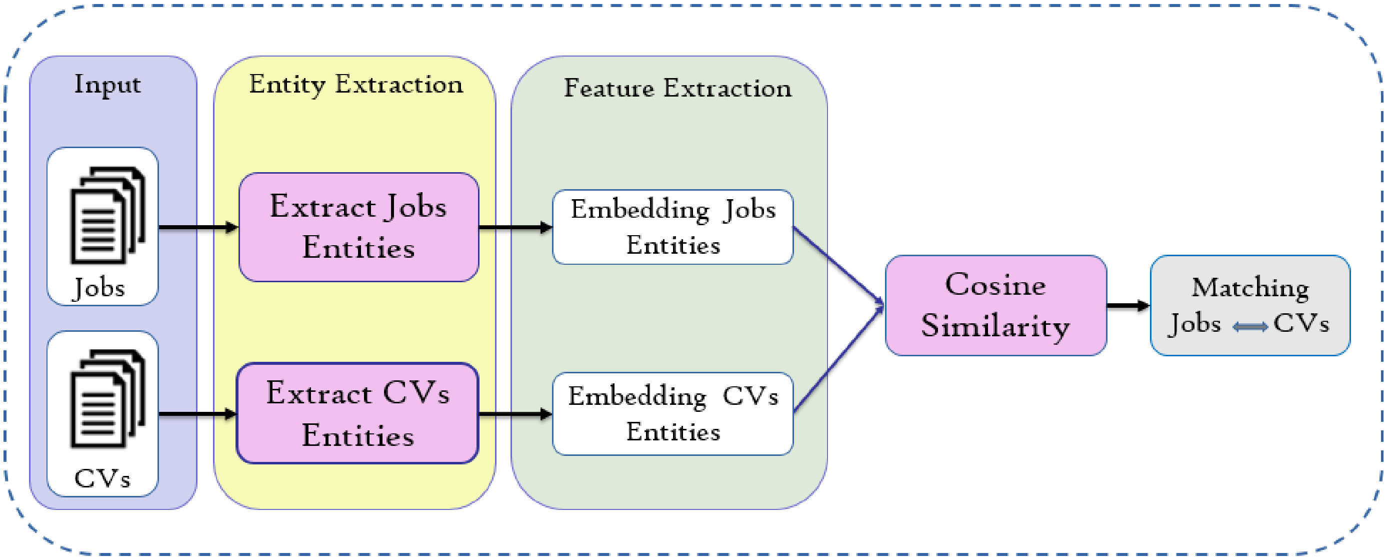

Fig. 2. Overview of the matching framework.

from both the job descriptions and the CVs. These extracted entities are then passed to a feature extraction module, which employs a pre-trained transformer model to produce contextual embeddings. Finally, cosine similarity is computed to assess the semantic alignment between each job description and the resumes. If the similarity score exceeds a given threshold, the corresponding CV is considered a potential match.

1) Input: The system takes as input a job description $J = \{J_1, J_2, \dots, J_n\}$ and a set of resumes $R = \{R_1, R_2, \dots, R_n\}$, where each J_i and R_i consists of unstructured text written in French. The input for each comparison is formally defined as Input $= (J_i, R_i)$, representing the pairing between a job description and a candidate resume.

2) Entity extraction: Entities such as skills, technologies, and job-related information are extracted from both the job description and the resumes using a Named Entity Recognition (NER) model F. For each pair, the model is applied as $S_{J_i} = F(J_i)$ and $S_{R_i} = F(R_i)$, where S_{J_i} denotes the set of entities extracted from the job description J_i, and S_{R_i} corresponds to the entities extracted from the i-th resume R_i, for all $i \in \{1, \dots, n\}$.

3) Feature extraction: The extracted entities are then passed through a CamemBERT-based transformer model which serves as the embedding function Em to generate dense vector representations [see Eq. (3)]:

$$U = Em(S_{J_i}), \quad V_i = Em(S_{R_i}) \quad \forall i \tag{3}$$

where, $U \in \mathbb{R}^d$ is the embedding vector for the job description, and $V_i \in \mathbb{R}^d$ represents the embedding vector for the i-th resume.

4) Similarity: To measure the alignment between a job description and a resume, cosine similarity is computed between their corresponding embeddings, as defined in Eq. (4):

$$\text{sim}(U, V_i) = \frac{U \cdot V_i}{\|U\| \cdot \|V_i\|}, \quad \forall i \tag{4}$$

A resume is considered a match if the similarity in Eq. (5) exceeds a predefined threshold:

$$\text{sim}(U, V_i) \geq \tau \tag{5}$$

where, $\tau = 0.5$ denotes the similarity threshold.

This similarity-based matching approach between candidates and job descriptions, as formulated in Eq. (5), enables a more accurate and efficient recruitment process.

C. Entity-Based Question Generation

After a match is confirmed between candidates and job descriptions, the process triggers a Large Language Model to generate relevant interview questions. For this task, we evaluated two LLMs: Google Gemini and Meta's LLaMA. While both models are capable of producing grammatically correct and contextually appropriate questions, Google Gemini demonstrated superior performance in terms of relevance and alignment with the extracted entities.

Gemini's advantages lie in its advanced instruction tuning, stronger contextual reasoning, and ability to generate coherent and targeted questions without the need for additional fine-tuning. On the other hand, LLaMA-based models often require instruction-tuned variants to achieve comparable results.

Therefore, Google Gemini is preferred in our pipeline due to its robustness, high-quality output, and ease of integration.

V. Experimental Results and Evaluation

We assessed in this work several approaches and models for entity extraction and matching between candidates and job descriptions.

A. Extract Entities

In this subsection, we first introduce the dataset used in the experiments. We then describe the experimental setup for both NER-based entity extraction model and the baseline models. Finally, we present a series of experimental results to validate the effectiveness of the proposed model.

1) Data description: We performed entity extraction to improve hiring efficiency in recruitment systems by applying Named Entity Recognition to the task of parsing and analyzing resumes. To support this, we compiled a dataset comprising both structured and semi-structured resume documents. The dataset, stored in .json format, included a total of 1,000 resumes. Of these, 800 were used for training and 200 for testing. This 80/20 split was adopted to ensure a reliable assessment of the model's ability to generalize to unseen data.

Each resume was annotated with key entities relevant to recruitment and candidate evaluation. These included: Name, LinkedIn Link, Location, Degree, College Name, Email Address, University Name, Skills and Certifications. These entities were carefully chosen to reflect the most critical information recruiters seek during the hiring process. The dataset covered a broad range of resume formats and writing styles, thereby improving the robustness and generalization capabilities of the NER models.

2) Implementation setup: We evaluate the proposed entity extraction method to improve hiring efficiency in recruitment systems by applying Named Entity Recognition to the dataset described earlier. This dataset provides a solid foundation for assessing the performance of various text-based models under different hyperparameter configurations.

Our model was trained using the hyperparameters in Table I. Specifically, we used RoBERTa as our underlying transformer for feature extraction due to its superior performance on language understanding tasks. 0.1 was the dropout rate that we utilized in order to reduce the probability of overfitting by randomly eliminating neurons when training. We employed the Adam optimizer, where learning rates are adapted dynamically for all parameters, and set the initial learning rate to 5×10^{-5} in order to achieve steady and smooth convergence.

TABLE I. SUMMARY OF MODEL HYPERPARAMETERS AND DETAILS

Category	Value
Evaluation Metrics	F1-Score, Precision, Recall
Feature Extraction	RoBERTa
Dropout Rate	0.1
Optimizer	Adam
Learning Rate	5×10^{-5}
Early Stopping	1600 steps
GPU Allocator	PyTorch
Max Steps	20

For monitoring training and preventing overfitting, we employed early stopping with patience 1600 steps, i.e., training would be halted if improvement in validation loss was not observed within that time. Training was conducted for a maximum of 20 steps (or epochs), with adequate iterations for the model to learn without incurring high computational expenses. Training was conducted in PyTorch with GPU acceleration, providing efficient use of resources through dynamic memory allocation and fast batch processing.

For evaluating the performance of the model, we employed typical NER evaluation metrics: Precision, which provides an estimate of the accuracy of the entity predictions; Recall, which provides an estimate of the capability of the model to capture all entities of interest; and F1-score, the harmonic mean of recall and precision. These estimates provide an overall estimate of the capability of the model to correctly and comprehensively extract entity information from resumes.

3) Model variations: To assess the performance of various transformer-based architectures for Named Entity Recognition (NER) in the context of resume entity extraction, experiments were conducted using three widely adopted pre-trained models: BERT, DeBERTa, and RoBERTa, each employed as a feature extractor. All models were fine-tuned under identical training conditions to ensure a fair comparison. Their performance was evaluated across multiple entity types using standard evaluation metrics, including Precision, Recall, and F1-score.

Table II presents the relative performance of the three models across ten target entity types. All results are listed in percentages (%) for each measurement. RoBERTa generally outperformed both BERT and DeBERTa for nearly all categories, achieving the highest F1-scores in major domains such as Name, LinkedIn Link, Location, Degree, College Name, Email Address, University Name, Skills and Certifications, and also in the overall score for all entities

RoBERTa achieved a combined F1-score of 85.11%, outperforming DeBERTa (82.94%) and BERT (80.14%). This improvement is a witness to the enhanced contextual depth understanding capacity of RoBERTa for resume content. The model performed strongly in proper categories like Email Address and Skills, which are central to hiring decision-making.

While BERT was a decent baseline, it lagged behind in recall for many entity types, indicating a bias towards missing relevant entities. DeBERTa had better recall than BERT, especially in categories such as Email Address and Location, but was less dependable overall than RoBERTa.

These findings confirm that choice of transformer architecture exerts a significant impact on the effectiveness of NER-based resume parsing. Of the models tested, RoBERTa emerged as the most powerful and most reliable and therefore an infinitely feasible option to use in automated hiring processes.

B. Matching Evaluation

In this part of our analysis, we evaluate the performance of the candidates–jobs matching component. We begin by introducing the similarity-based scoring framework used to quantify the relevance between candidates and recruiters. We then present a summary of the matching performance, including evaluation metrics and comparative analysis.

1) Similarity-based scoring framework: To assess the effectiveness of our matching system, we use a similarity-based scoring approach. When a CV or a job description is entered,

TABLE II. Comparison of Our NER Results on Resumes (%)

Entities	Feature Extraction								
	BERT			DeBERTa			RoBERTa		
	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall
All Entities	80.14	80.43	81.91	82.94	83.61	83.37	**85.11**	**85.24**	**86.47**
Name	79.09	79.34	80.57	81.65	83.86	79.66	**85.20**	**85.06**	**87.56**
LinkedIn Link	78.95	79.31	79.33	77.52	83.95	71.73	**84.75**	**84.71**	**87.79**
Location	79.24	78.17	80.34	81.91	83.14	82.28	**84.69**	**84.86**	**86.47**
Degree	78.49	77.67	79.46	81.44	83.26	80.52	**85.28**	**84.47**	**87.02**
College Name	78.81	78.14	79.50	78.09	83.12	74.83	**84.82**	**84.36**	**85.88**
Email Address	80.37	79.10	81.70	84.85	84.11	85.59	**85.54**	**85.63**	**87.33**
University Name	78.92	77.43	80.46	81.52	83.10	80.00	**84.90**	**85.19**	**85.78**
Skills	80.01	79.89	80.57	84.22	84.94	84.65	**85.03**	**84.94**	**86.83**
Certifications	78.04	76.57	79.58	78.90	83.52	75.60	**85.20**	**84.82**	**87.08**

the system calculates a semantic similarity score between them using embeddings generated by the CamemBERT model. The evaluation pipeline is detailed below:

a) Entity embedding for job description and CVs: Let T denote the job description text and C_i represent the entities (e.g., skills) extracted from the i-th CV. The embeddings for both are obtained via the CamemBERT model, as defined in Eq. (6):

$$E_T = \mathrm{E}_m(T), \quad E_{C_i} = \mathrm{E}_m(\text{Entities}_i) \tag{6}$$

where, T denotes the job description text, Entities_i represents the set of extracted skills from the i-th candidate's CV, E_T is the embedding vector corresponding to the job description, and E_{C_i} refers to the embedding of the entities extracted from the i-th CV.

b) Similarity computation using cosine metric: The similarity score KM_i between the job description and each CV is calculated using cosine similarity [Eq. (7)]:

$$\text{KM}_i = \frac{E_T \cdot E_{C_i}}{\|E_T\|\|E_{C_i}\|} \tag{7}$$

where, $E_T \cdot E_{C_i}$ denotes the dot product between the job description and the candidate CV embeddings, and $\|E_T\|$ and $\|E_{C_i}\|$ are the Euclidean norms of the respective embedding vectors.

c) Score normalization: To ensure consistent scoring across different inputs, the similarity scores are normalized using their mean μ and standard deviation σ, as given in Eq. (8):

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \text{KM}_i, \quad \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\text{KM}_i - \mu)^2} \tag{8}$$

If $\sigma = 0$, it is replaced with $\sigma = 1$ to avoid division by zero. The normalized difference is then computed [Eq. (9)]:

$$\text{Diff}_i = \frac{\text{KM}_i - \mu}{\sigma} \tag{9}$$

d) Rating conversion: The normalized score is mapped on a 0 to 10 rating scale using the transformation in Eq. (10):

$$\text{Rating}_i = \min\left(10, \max\left(0, 5 + \text{Diff}_i\right)\right) \tag{10}$$

This ensures that:

- Ratings remain within the range [0, 10].
- Ratings are centered around a neutral baseline of 5.

e) Matching accuracy metric: Finally, the matching accuracy is computed as the average rating of the top k most similar CVs, as shown in Eq. (11):

$$\text{Accuracy} = \frac{1}{k}\sum_{i=1}^{k} \text{Rating}_i \tag{11}$$

In our evaluation, $k = 5$ was used.

2) Summary of matching results: The matching accuracy was evaluated using the method described in the previous section (Accuracy Computation Method). The result reflects the average performance of the top five matched CVs, based on cosine similarity between job descriptions and resume embeddings.

The evaluation shows that the CamemBERT-based model achieves strong performance, with an accuracy of 92%. This highlights the effectiveness of the proposed semantic matching pipeline in retrieving relevant candidate profiles from unstructured textual data.

These findings suggest that the proposed AI-based job matching system offers accurate candidate recommendations, efficient processing, and strong potential for deployment in real-world recruitment workflows.

VI. Conclusion and Future Work

In this work, we propose an approach that automates and optimizes the recruitment process. By combining entity extraction, candidates–jobs matching, and entity-based question generation, the approach offers a complete solution for streamlining the recruitment process. Leveraging natural language processing techniques and semantic similarity models, the process reduces manual effort, enhances candidate evaluation, and increases objectivity in decision making. One major challenge encountered is the handling of resumes that contain text embedded in images, often the case with visually designed resumes. This can lead to data loss and the unintended exclusion of qualified candidates. Addressing this limitation is crucial to ensure data integrity and maintain the system's effectiveness across varied recruitment contexts.

Future work could focus on incorporating OCR (Optical Character Recognition) technologies or image-to-text conversion models to enable the processing of a broader range of resume formats. Enhancing the system with real-time feedback mechanisms, improving its scalability for large-scale hiring scenarios, and expanding its adaptability across different industries would further increase its utility. Moreover, integrating context-aware decision-making and optimizing computational performance could elevate the overall intelligence and responsiveness of the system. These developments would contribute to building a more resilient and inclusive recruitment solution, delivering tangible benefits for both organizations and job seekers.

References

[1] A. Sharma, Amrita, S. Chakraborty, and S. Kumar, "Named entity recognition in natural language processing: A systematic review," in *Proceedings of Second Doctoral Symposium on Computational Intelligence: DoSCI 2021*, Springer, 2022, pp. 817–828. https://doi.org/10.1007/978-981-16-3346-1_66https://doi.org/10.1007/978-981-16-3346-1_66.

[2] Mansouri, A., Affendey, L. S., & Mamat, A. (2008). "Named entity recognition approaches." *International Journal of Computer Science and Network Security*, 8(2), 339–344. Citeseer. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=f3a467f1a32b844b3f04321212d49ee4f215d484doi: f3a467f1a32b844b3f04321212d49ee4f215d484.

[3] Li, J., Sun, A., Han, J., & Li, C. (2020). "A survey on deep learning for named entity recognition." *IEEE Transactions on Knowledge and Data Engineering*, 34(1), 50–70. IEEE. https://doi.org/10.1109/TKDE.2020.2981314https://doi.org/10.1109/TKDE.2020.2981314.

[4] Vychegzhanin, S., & Kotelnikov, E., "Comparison of Named Entity Recognition Tools Applied to News Articles," *Proceedings of the 2019 Ivannikov ISPRAS Open Conference (ISPRAS)*, pp. 72–77, 2019, IEEE. https://doi.org/10.1109/ISPRAS47671.2019.00017https://doi.org/10.1109/ISPRAS47671.2019.00017.

[5] P. Sun, X. Yang, X. Zhao, and Z. Wang, "An overview of named entity recognition," in *2018 International Conference on Asian Language Processing (IALP)*, IEEE, 2018, pp. 273–278. https://doi.org/10.1109/IALP.2018.8629225https://doi.org/10.1109/IALP.2018.8629225.

[6] X. Schmitt, S. Kubler, J. Robert, M. Papadakis, and Y. LeTraon, "A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, IEEE, 2019, pp. 338–343. https://doi.org/10.1109/SNAMS.2019.8931850https://doi.org/10.1109/SNAMS.2019.8931850.

[7] H. Shelar, G. Kaur, N. Heda, and P. Agrawal, "Named entity recognition approaches and their comparison for custom ner model," *Science & Technology Libraries*, vol. 39, no. 3, pp. 324–337, 2020, Taylor & Francis. https://doi.org/10.1080/0194262X.2020.1759479https://doi.org/10.1080/0194262X.2020.1759479.

[8] C. Gayathri and R. S. Ravindran, "Named entity recognition using Bi-LSTM model with pointer cascade conditional random field for selecting high-profit products," *Egyptian Informatics Journal*, vol. 31, p. 100703, 2025. https://doi.org/10.1016/j.eij.2025.100703

[9] Y. Hu, Y. Chen, and Y. Xu, "A shape composition method for named entity recognition," *Neural Networks*, vol. 187, p. 107389, 2025. https://doi.org/10.1016/j.neunet.2025.107389

[10] A. Thukral, S. Dhiman, R. Meher, and P. Bedi, "Knowledge graph enrichment from clinical narratives using NLP, NER, and biomedical ontologies for healthcare applications," *International Journal of Information Technology*, vol. 15, no. 1, pp. 53–65, 2023, Springer. https://doi.org/10.1007/s41870-022-01145-yhttps://doi.org/10.1007/s41870-022-01145-y.

[11] Chantrapornchai, C., & Tunsakul, A., "Information Extraction Based on Named Entity for Tourism Corpus," *Proceedings of the 2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 187–192, 2019, IEEE. https://doi.org/10.1109/JCSSE.2019.8864166https://doi.org/10.1109/JCSSE.2019.8864166.

[12] S. Pudasaini, S. Shakya, S. Lamichhane, S. Adhikari, A. Tamang, and S. Adhikari, "Application of NLP for information extraction from unstructured documents," in *Expert Clouds and Applications: Proceedings of ICOECA 2021*, Springer, 2022, pp. 695–704. https://doi.org/10.1007/978-981-16-2126-0_54https://doi.org/10.1007/978-981-16-2126-0_54.

[13] Q. Zeng, M. Yuan, Y. Su, J. Mi, Q. Che, and J. Wan, "Improving Multimodal Named Entity Recognition via Text-image Relevance Prediction with Large Language Models," *Neurocomputing*, p. 130982, 2025. https://doi.org/10.1016/j.neucom.2025.130982

[14] Z. Zhou, L. Wei, and H. Luan, "Deep learning for named entity recognition in extracting critical information from struck-by accidents in construction," *Automation in Construction*, vol. 173, p. 106106, 2025. https://doi.org/10.1016/j.autcon.2025.106106

[15] X. Guo, Y. Chen, R. Tang, and Q. Zheng, "Camouflaged named entity recognition in 2D sentence representation," *Expert Systems with Applications*, vol. 257, p. 125096, 2024. https://doi.org/10.1016/j.eswa.2024.125096

[16] Guo, Shiqiang, Alamudun, Folami, & Hammond, Tracy, "RéSuMatcher: A Personalized Résumé-Job Matching System," *Expert Systems with Applications*, vol. 60, pp. 169–182, 2016, Elsevier. https://doi.org/10.1016/j.eswa.2016.04.013https://doi.org/10.1016/j.eswa.2016.04.013.

[17] Zaroor, A., Maree, M., & Sabha, M., "JRC: A Job Post and Resume Classification System for Online Recruitment," *Proceedings of the 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 780–787, 2017, IEEE. https://doi.org/10.1109/ICTAI.2017.00123https://doi.org/10.1109/ICTAI.2017.00123.

[18] Zaroor, A., Maree, M., & Sabha, M., "A Hybrid Approach to Conceptual Classification and Ranking of Resumes and Their Corresponding Job Posts," *Proceedings of the 9th KES International Conference on Intelligent Decision Technologies (KES-IDT 2017)–Part I*, pp. 107–119, 2018, Springer. https://doi.org/10.1007/978-3-319-59421-7_10https://doi.org/10.1007/978-3-319-59421-7_10.

[19] Martinez-Gil, J., Paoletti, A. L., & Pichler, M., "A Novel Approach for Learning How to Automatically Match Job Offers and Candidate Profiles," *Information Systems Frontiers*, vol. 22, pp. 1265–1274, 2020, Springer. https://doi.org/10.1007/s10796-019-09929-7https://doi.org/10.1007/s10796-019-09929-7.

[20] Sarveshwaran, R., Karthikeyan, S., Cruz, Meenalosini V., Shreyanth, S., Niveditha, S., & Rajesh, P.K., "NLP-Based AI-Driven Resume Screening Solution for Efficient Candidate Selection," *International Congress on Information and Communication Technology*, pp. 359–370, 2024, Springer. https://doi.org/10.1007/978-981-97-3556-3_29https://doi.org/10.1007/978-981-97-3556-3_29.

[21] A. Khatua and W. Nejdl, "Matching recruiters and jobseekers on Twitter," in *Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2020, pp. 266–269. https://doi.org/10.1109/ASONAM49781.2020.9381392

[22] Y. Rebboud, L. Tailhardat, P. Lisena, and R. Troncy, "Can LLMs Generate Competency Questions?," in *Proceedings of the European Semantic Web Conference (ESWC)*, Springer, 2024, pp. 71–80. [Online]. Available: https://doi.org/10.1007/978-3-031-78952-6_7

[23] A. Gendrin, L. Souliotis, J. Loudon-Griffiths, R. Aggarwal, D. Amoako, G. Desouza, S. Dimitrievska, P. Metcalfe, E. Louvet, and H. Sahni, "Identifying patient populations in texts describing drug approvals through deep learning–based information extraction: Development of a natural language processing algorithm," *JMIR Formative Research*, vol. 7, p. e44876, 2023. https://doi.org/10.2196/44876

[24] Al-Moslmi, T., Gallofré Ocaña, M., Opdahl, A. L., and Veres, C., "Named entity extraction for knowledge graphs: A literature overview," *IEEE Access*, vol. 8, pp. 32862–32881, 2020. https://doi.org/10.1109/ACCESS.2020.2973928https://doi.org/10.1109/ACCESS.2020.2973928.

[25] Pham, V. L., Vu, N. S., & Nguyen, V. V. (2018). "Study of information extraction in resume." In Conference Proceedings. https://eprints.uet.vnu.edu.vn/eprints/id/eprint/3349/https://eprints.uet.vnu.edu.vn/eprints/id/eprint/3349/

[26] C. Zhu, X. Hu, H. Wu, C. Qin, H. Zhu, and H. Xiong, "Enhancing job recommendations with LLM-based resume completion: A behavior-denoised alignment approach," *Information Processing & Management*, vol. 62, no. 6, p. 104261, 2025. https://doi.org/10.1016/j.ipm.2025.104261

[27] Vanetik, N.; Kogan, G. "Job vacancy ranking with sentence embeddings, keywords, and named entities." *Information* **2023**, *14*, 468. https://doi.org/10.3390/info14080468https://doi.org/10.3390/info14080468.

[28] Bakliwal, A.; Gandhi, S. M.; Haribhakta, Y. "Leveraging Knowledge Graphs for Orphan Entity Allocation in Resume Processing." In *Proceedings of the 2023 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*; IEEE: 2023; pp. 123–128. https://doi.org/10.1109/IICAIET59451.2023.10291293https://doi.org/10.1109/IICAIET59451.2023.10291293.

[29] Wang, Yu, Yining Sun, Zuchang Ma, Lisheng Gao, Yang Xu, and Ting Sun. "Application of pre-training models in named entity recognition." In *Proceedings of the 2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 1, pp. 23-26. IEEE, 2020. https://doi.org/10.1109/IHMSC49165.2020.00013https://doi.org/10.1109/IHMSC49165.2020.00013.

[30] Tran, Viet-Trung, Hai-Nam Cao, and Tuan-Dung Cao. "A practical method for occupational skills detection in Vietnamese job listings." In *Proceedings of the Asian Conference on Intelligent Information and Database Systems*, pp. 571-581. Springer, 2022. https://doi.org/10.1007/978-3-031-21743-2_46https://doi.org/10.1007/978-3-031-21743-2_46.

[31] Rosenberger, Julian, Lukas Wolfrum, Sven Weinzierl, Mathias Kraus, and Patrick Zschech. "CareerBERT: Matching resumes to ESCO jobs in a shared embedding space for generic job recommendations." *Expert Systems with Applications*, Elsevier, 2025, p. 127043. https://doi.org/10.1016/j.eswa.2025.127043https://doi.org/10.1016/j.eswa.2025.127043.

[32] Tyagi, Swati, Wei Qian, Jiaheng Xie, Rick Andrews, and others. "Enhancing gender equity in resume job matching via debiasing-assisted deep generative model and gender-weighted sampling." *International Journal of Information Management Data Insights*, vol. 4, no. 2, Elsevier, 2024, pp. 100283. https://doi.org/10.1016/j.jjimei.2024.100283https://doi.org/10.1016/j.jjimei.2024.100283.

[33] A. Barducci, S. Iannaccone, V. La Gatta, V. Moscato, G. Sperlì, and S. Zavota, "An end-to-end framework for information extraction from Italian resumes," *Expert Systems with Applications*, vol. 210, p. 118487, 2022. https://doi.org/10.1016/j.eswa.2022.118487

[34] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, vol. 1, pp. 4171–4186, Minneapolis, Minnesota, Jun. 2019. https://doi.org/10.18653/v1/N19-1423