Exploring the Impact of Annotation Schemes on Arabic Named Entity Recognition across General and Specific Domains

Taoufiq El Moussaoui
LISAC Laboratory, Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco
taoufiq.elmoussaoui@usmba.ac.ma (corresponding author)

Chakir Loqman
LISAC Laboratory, Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco
loqman.chakir@usmba.ac.ma

Jaouad Boumhidi
LISAC Laboratory, Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco
jaouad.boumhidi@usmba.ac.ma

Received: 11 January 2025 | Revised: 11 February 2025, 15 February 2025, and 18 February 2025 | Accepted: 21 February 2025

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | *DOI: https://doi.org/10.48084/etasr.10205*

ABSTRACT

Named Entity Recognition (NER) is a fundamental task in natural language processing (NLP) that involves identifying and classifying entities into predefined categories. Despite its importance, the impact of annotation schemes and their interaction with domain types on NER performance, particularly for Arabic, remains underexplored. This study examines the influence of seven annotation schemes (IO, BIO, IOE, BIOES, BI, IE, and BIES) on arabic NER performance using the general-domain ANERCorp dataset and a domain-specific Moroccan legal corpus. Three models were evaluated: Logistic Regression (LR), Conditional Random Fields (CRF), and the transformer-based Arabic Bidirectional Encoder Representations from Transformers (AraBERT) model. Results show that the impact of annotation schemes on performance is independent of domain type. Traditional Machine Learning (ML) models such as LR and CRF perform best with simpler annotation schemes like IO due to their computational efficiency and balanced precision-recall metrics. On the other hand, AraBERT excels with more complex schemes (BIOES, BIES), achieving superior performance in tasks requiring nuanced contextual understanding and intricate entity relationships, though at the cost of higher computational demands and execution time. These findings underscore the trade-offs between annotation scheme complexity and computational requirements, offering valuable insights for designing NER systems tailored to both general and domain-specific Arabic NLP applications.

Keywords-Arabic NER; annotation schemes; general-domain NER; domain-specific NER; AraBERT

I. INTRODUCTION

Named Entity Recognition (NER) is a core task in Natural Language Processing (NLP), aiming to identify and classify the entities in a text into predefined categories [1]. Given a sequence of terms, $Seq =< t_1, t_2, .., t_N >$, the goal of NER is to produce a list of tuples $< Ind_s, Ind_e, NE >$, where Ind_s and Ind_e denote the start and end indexes of the entity, and NE represents its type. These Named Entities (NEs) can be categorized either into general entities, common across domains, or domain-specific entities, specialized in fields like law or medicine [1]. Figure 1 presents an example of the NER task.

Annotation schemes define how tokens in a dataset are labeled. This study investigates seven widely used annotation schemes and evaluates their impact on three Arabic NER models. These schemes are:

- **IO**: A simple scheme with two tags: "I" for tokens in a NE and "O" for non-entities. Its main limitation is in distinguishing consecutive entities of the same type. For instance, in the sentence: الرباط، فاس (Rabat, Fez), both الرباط،

(Rabat,) and فاس (Fez) would be labeled as a single entity, despite representing distinct locations.

- **BIO** (or IOB): Introduces a "B" tag to mark the beginning of entities, addressing IO's inability to separate consecutive entities. Adopted by the Conference on Computational Natural Language Learning (CoNLL), BIO is a standard in NER for effectively distinguishing entity boundaries [2].
- **IOE**: Replaces BIO's "B" tag with "E" to mark entity endings instead of beginnings.
- **BIOES**: Extends BIO and IOE by adding the "S" tag for single-token entities. Equivalent to the BILOU scheme, it uses "E" instead of "L" (last) and "S" instead of "U" (unique).
- **BI**: Similar to BIO but labels non-entity tokens as "B-O" (beginning) and "I-O" (other positions).
- **IE**: A variant of IOE, marking the end of non-entity with "E-O" and labeling other non-entity tokens as "I-O".
- **BIES**: Extends BIOES by labeling non-entity tokens with "B-O" for the beginning, "I-O" for the inside, "E-O" for the end, and "S-O" for single-token non-entity sequences.

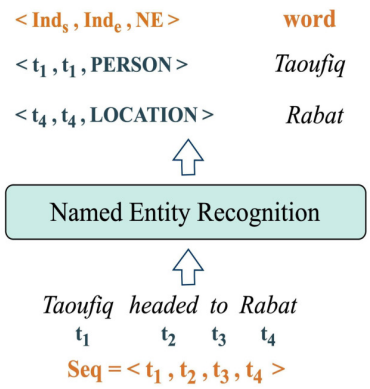

Fig. 1. Example of the NER task.

The performance of NER models is significantly influenced by the annotation schemes used for training data. However, the impact of these schemes on Arabic NER models, especially with datasets containing both general and domain-specific NEs, remains underexplored. Identifying the most effective scheme based on NE types is crucial for improving accuracy and efficiency. Several studies have addressed the selection of annotation schemes.

Authors in [3] evaluated seven annotation schemes (IO, BIO, IOE, BIOES, BI, IE, BIES) on the performance of five ML models, including Adaptive Boosting (AdaBoost), decision tree, K-Nearest Neighbors (KNN), Random Forest (RF), and gradient boost. Using a dataset of 27 Arabic medical articles, they reported that the simple IO scheme achieved the highest f-measure score of 84.44%. However, it struggled with recognizing consecutive entities, a capability offered by more complex schemes like BIES. Authors in [4] evaluated the IO, BIO, IOE, IOBE, IOBS, IOES, BIOES schemes using CRF, Multinomial Naive Bayes (MNB), and Support Vector Machine (SVM) classifiers on the Arabic ANERCorp dataset. It found that the IO scheme outperformed others in precision, recall, and F-measure. Authors in [5] introduced the BIL2 scheme for NER in Urdu, showing its superiority with Hidden Markov Models (HMM) and CRF classifiers. Authors in [6] found that IOE-1 and IOE-2 schemes outperformed others using CRF and Maximum Entropy models across multiple languages. Authors in [7] demonstrated that BILOU outperformed BIO with CRF, achieving an F-measure of 87% for Estonian NER. Authors in [8] showed that IO outperformed BILOU for CRF-based NER on the HAREM corpus.

While these studies highlight the critical role of annotation schemes, most research overlooks comparisons with Deep Learning (DL) models and the effect of NE types (general vs. domain-specific) on performance. This study addresses this gap by analyzing the impact of annotation schemes on Arabic NER models across different NE domains. We also evaluate the execution time complexity for each scheme, providing a holistic view of the trade-offs involved.

In this paper, we contribute to Arabic NER by examining how different annotation schemes affect performance across both generic and domain-specific contexts. Specifically, we use the ANERCorp dataset for general-domain NEs and construct a dedicated Moroccan legal corpus for domain-specific entities. To systematically assess the impact of annotation strategies, we apply seven different annotation schemes to create multiple dataset versions. We then evaluate these schemes using two ML models, namely LR and CRF, alongside the DL-based AraBERT transformer model. Furthermore, we analyze the execution time complexity associated with each annotation scheme, providing valuable insights into their computational efficiency.

II. RESEARCH METHODOLOGY

This section provides an overview of the datasets used, the process of preparing these datasets with different annotations, and the models employed in the experiments.

A. Datasets

To investigate the impact of annotation schemes and data domains on arabic NER, we utilized two datasets.

1) ANERCorp

ANERCorp is a publicly available Arabic corpus derived from 316 articles across various newspapers [9]. It contains 165,535 NEs and 148,252 tokens, serving as a general-domain dataset for Arabic NER. The corpus includes four NE categories: PERS (person), LOC (location), ORG (organization), and MISC (miscellaneous, for entities not fitting other categories), with non-entity words labeled as O. Annotated using the BIO tagging scheme, the tags include O, B-PERS, I-PERS, B-LOC, I-LOC, B-ORG, I-ORG, B-MISC, and I-MISC. Table I presents the distribution of NEs within the corpus.

TABLE I. DISTRIBUTION OF NAMED ENTITIES IN THE ANERCORP CORPUS

Entity	Percentage (%)
PERS	39.0
LOC	30.4
ORG	20.6
MISC	10.0

2) Moroccan Legal Corpus

The Moroccan legal corpus was created to support our study by focusing on domain-specific entities. It consists of 632 Arabic judgments issued by Moroccan courts, specifically related to criminal cases between 2015 and 2019. The corpus contains 31,051 entities and 101,029 tokens and was annotated using the BIO tagging scheme. It includes eleven NEs: PERS (person's name), LOC (location), CIN (national identity card), BIRTH_DATE (birth date), FAMILY_STAT (marital status), PROF (profession), JUDG_DATE (judgment date), CASE_NUM (case number), CHARGE (the charge), PEN (penalty), JUDG (judgment), and O (for non-entity words). Figure 2 illustrates the process of creating the Moroccan legal corpus, while Table II presents the distribution of named entities within the corpus.

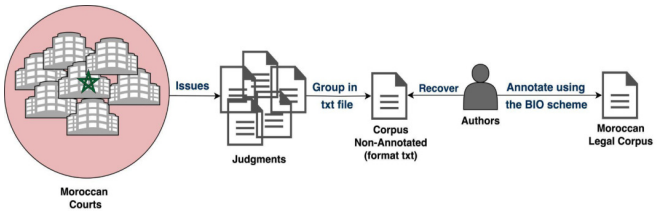

Fig. 2. The process of creating the Moroccan legal corpus.

TABLE II. DISTRIBUTION OF NAMED ENTITIES IN THE MOROCCAN LEGAL CORPUS

Entity	Percentage (%)
PERS	18.6
LOC	24.3
CIN	1.0
FAMILY_STAT	2.3
BIRTH_DATE	2.3
PROF	2.6
CASE_NUM	2.0
JUDG_DATE	2.0
PEN	12.5
CHARGE	20.0
JUDG	12.4

Both datasets follow the CoNLL format, which consists of a two-column structure. The first column represents the words, while the second column contains the corresponding tags.

B. Datasets Preparation

In this step, we developed a python script *converter.py* that takes a dataset annotated in the BIO format as input and generates six different versions, each adhering to a specific annotation scheme: IO, IOE, BIOES, BI, IE, and BIES. The script implements algorithms that convert a BIO-annotated dataset into the aforementioned schemes.

C. Models

In this study, we evaluate the performance of various models for the Arabic NER task. These include traditional ML algorithms, such as LR and CRF, as well as AraBERT, a transformer model designed specifically for Arabic text.

1) Logistic Regression

LR is a linear classifier used for binary or multi-class classification problems. For NER, it is applied in a sequence labeling setup, where each token in a sentence is classified into one of several possible entity tags. Given a feature set $x = [x_1, x_2, \ldots, x_n]$ for a token, the model predicts the probability of each possible class (entity type) y as:

$$P(y = c|x) = \frac{e^{w_c^T x}}{\sum_{c'} e^{w_{c'}^T x}} \quad (1)$$

where w_c is the weight vector for class c (entity type), x is the feature vector for a token, and the denominator is the normalization term, ensuring that the probabilities for all classes sum to 1. The model learns the weights w_c for each class by minimizing the negative log-likelihood of the observed tags over the training data:

$$L(w) = -\sum_{i=1}^{N} log(P(y_i|x_i)) \quad (2)$$

where N is the number of training samples. Optimization is typically performed using gradient descent or a variant, such as stochastic gradient descent.

Before training the LR model, the data were processed to extract features representing each token in a format suitable for the model. These features included a) the token itself, b) whether it was the first or last word in the sentence, c) whether it was numeric, d) its prefixes and suffixes (1-2 characters long), and e) the previous and next tokens in the sentence. For the first and last tokens, special markers <START> and <END> were used to handle edge cases.

These features were collected for every token in all sentences. The resulting features were vectorized using DictVectorizer, which transforms them into a sparse matrix representation suitable for input into the LR model. In our approach, the LR model was trained on 80% of the data and evaluated on the remaining 20%, with the maximum number of iterations set to 200.

2) Conditional Random Fields

CRF are widely used for sequence labeling tasks, including NER. Unlike LR, which treats each token independently, CRFs model the dependencies between neighboring labels, making them particularly effective for structured prediction tasks [10]. CRFs are undirected probabilistic graphical models utilized to predict the sequence of labels $y = [y_1, y_2, \ldots, y_n]$ given a sequence of observations $x = [x_1, x_2, \ldots, x_n]$. The conditional probability of a label sequence y is expressed as:

$$P(y|x) = \frac{\exp(\sum_{i=1}^{n} \sum_k \lambda_k f_k(y_i, x, i) + \sum_j \mu_j g_j(y_{i-1}, y_i, x, i))}{Z(x)} \quad (3)$$

where $f_k(y_i, x, i)$ represents the feature functions for token-level features, $g_j(y_{i-1}, y_i, x, i)$ is the transition feature function

for label dependencies, λ_k and μ_j are the model parameters, and $Z(x)$ represents the partition function normalizing probabilities over all possible label sequences. The model parameters λ_k and μ_j are learned by maximizing the conditional log-likelihood of the observed sequences in the training data.

Before training, we extracted token-level features for each word in a sentence. Feature extraction was performed using a function that takes a sentence as input and generates the feature set for a single token by considering its context within the sentence. This includes neighboring words, positional indicators, and morphological characteristics such as prefixes and suffixes. This token-wise feature extraction enables the CRF model to capture both token-specific and contextual information for label prediction. The CRF model was trained using the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) algorithm, with regularization parameters $c_1 = 0.1$ and $c_2 = 0.1$ to prevent overfitting. The training process was configured with a maximum of 100 iterations to ensure convergence. The dataset was split into an 80% training set and a 20% test set, consistent with the approach used for the LR model, ensuring a fair comparison of performance.

3) AraBERT

AraBERT is a transformer-based language model specifically designed for processing Arabic text [11]. It is based on the Bidirectional Encoder Representations from Transformers (BERT) architecture [12] and has been pre-trained on a large corpus of Arabic text, making it highly suitable for tasks like Arabic NER. To fine-tune AraBERT, we processed our dataset by tokenizing each sentence and mapping labels to tokenized outputs, leveraging the AutoTokenizer class from the hugging face transformers library. The dataset was split into training, evaluation, and testing subsets, maintaining an 80/10/10 split. Each token in a sentence was associated with its corresponding label, and padding tokens were assigned an ignore index to exclude them from the loss computation. The model was initialized using the BertForTokenClassification class, configured with the number of unique labels in the dataset. The optimizer used was AdamW, and the learning rate was adjusted dynamically through a linear scheduler with warm-up steps. The training process involved batch-wise updates using a cross-entropy loss function, while gradient clipping was employed to stabilize training.

III. EXPERIMENTATION SETUP AND RESULTS

In this section, we outline the experimental setup, discuss the metrics employed for model evaluation, present the results, and engage in a detailed discussion.

A. Experimental Setup

The training of the three models was conducted on a Google Colab instance powered by an NVIDIA Tesla T4 GPU. Built on the Volta architecture, the Tesla T4 features 2560 CUDA cores and 16 GB of GDDR6 memory, delivering robust computational capabilities to efficiently accelerate the training process.

B. Metrics

To evaluate the performance of the models developed, three key metrics were employed: precision (P), recall (R), and f-measure (F_1) [1]. For the ML models, namely LR and CRF, micro-averaged precision, recall, and f-measure were utilized. The micro-averaging approach aggregates the contributions of all classes to compute a single set of metrics, ensuring a balanced evaluation across all entity types. Conversely, for the AraBERT model, normal precision, recall, and f-measure metrics were used to assess the model's performance.

C. Results

This subsection is divided into two parts. The first part presents the results of the experiments on the domain-general corpus, while the second part presents the results of the experiments on the domain-specific corpus.

1) Results on the Domain-General Corpus

The results of the experiments studying the impact of the annotation scheme on the domain-general corpus ANERCorp are reported in terms of precision, recall, and f-measure, as detailed in Tables III, IV, and V. Additionally, the execution time analysis is summarized in Table VI.

TABLE III. PRECISION SCORE (%) ON THE ANERCORP DATASET FOR ALL MODELS AND ANNOTATION SCHEMES

Models	Annotation schemes						
	IO	**BIO**	**IOE**	**BIOES**	**BI**	**IE**	**BIES**
LR	90.04	88.47	87.37	87.11	87.99	87.49	86.57
CRF	89.00	88.00	87.00	85.00	88.00	88.00	84.00
AraBERT	88.24	87.24	89.24	89.47	90.09	89.27	90.50

TABLE IV. RECALL SCORE (%) ON THE ANERCORP DATASET FOR ALL MODELS AND ANNOTATION SCHEMES

Models	Annotation schemes						
	IO	**BIO**	**IOE**	**BIOES**	**BI**	**IE**	**BIES**
LR	69.72	60.31	60.81	52.44	61.07	62.62	53.28
CRF	72.00	67.00	65.00	61.00	67.00	66.00	59.00
AraBERT	89.67	89.59	89.12	91.32	89.59	90.19	91.44

TABLE V. F-MEASURE SCORE (%) ON THE ANERCORP DATASET FOR ALL MODELS AND ANNOTATION SCHEMES

Models	Annotation schemes						
	IO	**BIO**	**IOE**	**BIOES**	**BI**	**IE**	**BIES**
LR	77.68	70.44	70.69	63.54	70.88	72.11	63.94
CRF	79.00	75.00	74.00	70.00	75.00	75.00	68.00
Avg ML	78.34	72.72	72.34	66.77	72.94	73.55	65.97
AraBERT	88.95	88.40	89.18	90.39	89.84	89.73	90.74

TABLE VI. EXECUTION TIME (s) ON THE ANERCORP DATASET FOR ALL MODELS AND ANNOTATION SCHEMES

Models	Annotation schemes						
	IO	**BIO**	**IOE**	**BIOES**	**BI**	**IE**	**BIES**
LR	20	30	27	47	46	40	99
CRF	10	14	15	24	14	14	27
Avg ML	15	22	21	35.5	30	27	63
AraBERT	1878	1804	1880	1806	1805	1872	1803

2) Results on the Domain-Specific Corpus

The results of the experiments investigating the impact of annotation schemes on the domain-specific Moroccan legal corpus are presented in terms of precision, recall, and f-measure, as shown in Tables VII, VIII, and IX. Furthermore, the execution time analysis is summarized in Table X.

TABLE VII. PRECISION SCORE (%) ON THE MOROCCAN LEGAL CORPUS FOR ALL MODELS AND ANNOTATION SCHEMES

Models	Annotation schemes						
	IO	**BIO**	**IOE**	**BIOES**	**BI**	**IE**	**BIES**
LR	99.16	98.57	98.45	94.36	98.53	98.49	94.53
CRF	99.20	99.13	98.97	94.97	99.18	98.85	95.23
AraBERT	98.77	98.92	99.01	99.34	98.52	98.46	99.25

TABLE VIII. RECALL SCORE (%) ON THE MOROCCAN LEGAL CORPUS FOR ALL MODELS AND ANNOTATION SCHEMES

Models	Annotation schemes						
	IO	**BIO**	**IOE**	**BIOES**	**BI**	**IE**	**BIES**
LR	98.19	97.48	96.70	92.15	97.56	96.83	92.83
CRF	98.87	98.51	98.41	94.29	98.66	97.88	94.83
AraBERT	99.38	98.99	99.20	99.41	98.85	99.33	99.39

TABLE IX. F-MEASURE SCORE (%) ON THE MOROCCAN LEGAL CORPUS FOR ALL MODELS AND ANNOTATION SCHEMES

Models	Annotation schemes						
	IO	**BIO**	**IOE**	**BIOES**	**BI**	**IE**	**BIES**
LR	98.66	98.01	97.52	93.17	98.03	97.61	93.61
CRF	99.03	98.81	98.68	94.62	98.91	98.35	94.79
Avg ML	98.84	98.41	98.10	93.89	98.47	97.97	94.20
AraBERT	99.07	98.95	99.10	99.37	98.68	98.89	99.31

TABLE X. EXECUTION TIME (s) ON THE MOROCCAN LEGAL CORPUS FOR ALL MODELS AND ANNOTATION SCHEMES

Models	Annotation schemes						
	IO	**BIO**	**IOE**	**BIOES**	**BI**	**IE**	**BIES**
LR	6	9	10	16	14	13	17
CRF	8	14	14	26	14	15	31
Avg ML	7	11.5	12	21	14	14	24
AraBERT	310	310	311	310	311	314	310

D. Analysis and Discussion

In this subsection, we provide an analysis and discussion of the results obtained from evaluating the impact of annotation schemes on Arabic NER, focusing on both domain-general and domain-specific datasets.

1) Discussion of Results on the Domain-General Corpus

The results in Tables III, IV, and V show how the precision, recall, and f-measure for each model vary across the considered annotation schemes used in the ANERCorp dataset, a general-domain corpus with general NEs such as person, location, organization, miscellaneous, and non-entity (O). The choice of annotation scheme significantly impacts model performance, especially for traditional ML models like LR and CRF. For these models, simpler schemes like IO yield the highest precision scores, with LR reaching 90.04% and CRF 89.00%. In contrast, more detailed schemes such as BIOES and BIES result in a drop in precision, with LR and CRF scoring around 87.11% and 86.57% for LR, and 85.00% and 84.00% for CRF, respectively. This suggests that the simplicity of the IO scheme aligns better with the boundary-differentiation limitations of traditional models. In contrast, the transformer-based AraBERT model demonstrates the opposite trend. It excels with more complex annotation schemes, achieving its highest precision with BIES (90.50%), followed by BI (90.09%) and BIOES (89.47%). This indicates that richer annotation schemes, which provide more detailed entity boundaries, play to AraBERT's strengths. Given its ability to leverage deep contextual relationships, AraBERT benefits from the additional granularity provided by complex schemes, allowing it to effectively handle the nuances of various NEs within the ANERCorp dataset.

Regarding recall, simpler annotation schemes like IO allow ML models to achieve the best recall, with LR at 69.72% and CRF at 72.00%. However, more intricate schemes, such as BIOES (LR: 52.44%, CRF: 61.00%) and BIES (LR: 53.28%, CRF: 59.00%), lead to a noticeable decline in recall. AraBERT, however, maintains stable and high recall across all schemes, with the highest observed under the BIES scheme (91.44%), closely followed by BIOES (91.32%) and IE (90.19%). This stability highlights AraBERT's capacity to capture contextual dependencies effectively, regardless of the complexity of the annotation scheme.

The f-measure, which balances both precision and recall, shows a similar trend. Simpler schemes like IO lead to highest f-measures (LR: 77.68%, CRF: 79.00%). In contrast, more complex schemes like BIOES and BIES result in lower f-measure values, as they struggle to manage the increased boundary complexities. For AraBERT, however, the highest f-measures are achieved with the more detailed BIES (90.74%) and BIOES (90.39%), underlining how the model thrives with complex annotations. These results reinforce the idea that advanced models like AraBERT benefit from detailed annotations, which allow them to better resolve the complexities of entity boundaries in the general-domain context of ANERCorp. In summary, it can be said that simpler annotation schemes like IO and BIO are more suitable for traditional ML models, which are optimized for less complex boundary resolution. More intricate schemes, such as BIOES and BIES, enhance performance for transformer-based models like AraBERT, highlighting the importance of selecting the right annotation scheme based on the model's architecture. This analysis emphasizes that for effective NER in general-domain corpora like ANERCorp, the annotation scheme should be chosen in harmony with the model's capacity to handle complex entity structures.

When comparing average execution times (Table VI), ML models show a substantial advantage, with averages ranging from 15 s (IO) to 63 s (BIES), while DL models like AraBERT consistently require more than 1800 s across all annotation schemes. The complexity of the annotation scheme impacts execution time for ML models: simpler schemes such as IO lead to faster processing, while more complex schemes like BIOES and BIES result in higher computational load.

However, AraBERT's execution time remains uniformly high across all schemes, indicating that the model's performance is less affected by the complexity of the annotation scheme. These results underscore the trade-off between computational efficiency and model complexity, especially when selecting annotation schemes in resource-constrained environments.

The execution time results shown in Table VI reveal significant differences in computational demands across models and annotation schemes. Traditional ML models, such as LR and CRF, exhibit lower and more variable execution times compared to the computationally intensive AraBERT. LR demonstrates the highest variability in execution time, ranging from 20 (IO) to 99 s (BIES), while CRF maintains faster and more consistent times, ranging from 10 (IO) to 27 s (BIES). In contrast, AraBERT consistently requires significantly more time, ranging from 1803 (BIES) to 1880 s (IOE), reflecting the resource-intensive nature of transformer-based models.

2) Discussion of Results on the Domain-Specific Corpus

The results presented in Tables VII, VIII, and IX underscore the crucial role of annotation schemes in determining model performance on the Moroccan legal corpus. This corpus includes legal entities such as person, location, national identity card, birth date, marital status, profession, judgment date, case number, charge, penalty, judgment, and non-entity (O), where precise entity boundary detection is essential for accurate Arabic NER. Traditional ML models, including LR and CRF, perform best with the simpler IO annotation scheme, which treats each entity as a single, uninterrupted unit. These models achieve high precision (LR: 99.16%, CRF: 99.20%), as the simplicity of the scheme reduces ambiguity and allows for straightforward entity boundary identification. However, as annotation schemes become more complex (BIOES and BIES) and provide detailed positional information within entities, they experience a drop in precision. The added complexity challenges LR and CRF, as they struggle to accurately classify entities that span multiple tokens or contain intricate structures. In contrast, the transformer-based model AraBERT excels with more detailed annotation schemes. It achieves its highest precision with BIOES (99.34%) and BIES (99.25%), where each entity is assigned start and end markers as well as position information. This ability to leverage granular annotation data enables AraBERT to capture entities with more complex structures, particularly those that span multiple tokens, which is critical in legal texts where entities like charge and judgment may involve intricate relationships. The superior performance of AraBERT with these complex schemes highlights its capacity to learn contextual and structural dependencies, which traditional models struggle to capture.

Recall results reinforce the advantages of complex annotation schemes for transformer models. While traditional ML models such as LR and CRF show a decline in recall with more intricate schemes (e.g. BIOES and BIES), AraBERT maintains high and stable recall across all schemes, particularly under BIOES (99.41%). This consistency suggests that AraBERT is better equipped to handle the contextual relationships between entities, especially in complex legal texts, where boundaries may be ambiguous or span across multiple tokens.

Regarding f-measure (Table IX), AraBERT consistently outperforms LR and CRF across all annotation schemes. While the performance of LR and CRF peaks with the simpler IO scheme (LR: 98.66%, CRF: 99.03%), their f-measure declines with the more complex BIOES and BIES schemes. AraBERT, on the other hand, benefits from these detailed schemes, achieving its highest f-measure scores with BIOES (99.37%) and BIES (99.31%).

The execution time analysis further illustrates the trade-offs involved in using different annotation schemes (Table X). Traditional ML models like LR and CRF remain computationally efficient across all schemes, with execution times ranging from 6 to 31 s. As the complexity of the annotation schemes increases, particularly with BIES, these models exhibit a slight increase in execution time. However, the most significant computational cost is associated with AraBERT, which consistently requires more than 300 s to process the complex schemes. This stark contrast highlights the computational demands of transformer-based models, which must process intricate relationships and contextual information encoded by detailed annotation schemes.

IV. CONCLUSION

This paper explores how different annotation schemes (IO, BIO, IOE, BIOES, BI, IE, and BIES) affect Arabic Named Entity Recognition (NER) in both general and domain-specific datasets. The results indicate that traditional Machine Learning (ML) models, such as Logistic Regression (LR) and Conditional Random Fields (CRF), achieve optimal performance with simpler schemes like IO, showing high precision and recall in both general and specific contexts. However, their effectiveness diminishes with more complex schemes like BIOES and BIES. In contrast, the Arabic Bidirectional Encoder Representations from Transformers (AraBERT) surpasses these models when using complex annotation schemes, especially in the Moroccan legal corpus, where entity relationships are more complex. While traditional ML models are quicker and more reliable, AraBERT demands more resources but delivers superior accuracy. This study emphasizes the balance needed between model performance and computational efficiency when selecting the most optimal annotation scheme for Arabic NER tasks. Future research will aim to investigate hybrid models that combine the advantages of both traditional ML and transformer-based methods, as well as assess Arabic NER across a wider variety of domain-specific corpora to explore the applicability of annotation schemes in real-world scenarios.

REFERENCES

[1] T. E. Moussaoui and C. Loqman, "Advancements in Arabic Named Entity Recognition: A Comprehensive Review," *IEEE Access*, vol. 12, pp. 180238–180266, 2024, https://doi.org/10.1109/ACCESS.2024.3491897.

[2] E. F. T. K. Sang and S. Buchholz, "Introduction to the CoNLL-2000 Shared Task: Chunking," 2000, https://doi.org/10.48550/ARXIV.CS/0009008.

[3] N. Alshammari and S. Alanazi, "The impact of using different annotation schemes on named entity recognition," *Egyptian Informatics*

Journal, vol. 22, no. 3, pp. 295–302, Sep. 2021, https://doi.org/10.1016/j.eij.2020.10.004.

[4] I. Belhajem, "Effects of Multiple Annotation Schemes on Arabic Named Entity Recognition," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 17060–17067, Oct. 2024, https://doi.org/10.48084/etasr.8528.

[5] M. Kamran and S. Mansoor, "Named Entity Recognition System for Postpositional Languages: Urdu as a Case Study," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 10, 2016, https://doi.org/10.14569/IJACSA.2016.071019.

[6] M. Konkol and M. Konopík, "Segment Representations in Named Entity Recognition," in *Text, Speech, and Dialogue*, vol. 9302, P. Král and V. Matoušek, Eds. Cham: Springer International Publishing, 2015, pp. 61–70.

[7] A. Tkachenko, T. Petmanson, and S. Laur, "Named Entity Recognition in Estonian," in *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, Sofia, Bulgaria, Aug. 2013, pp. 78–83.

[8] D. O. F. do Amaral, M. Buffet, and R. Vieira, "Comparative Analysis between Notations to Classify Named Entities using Conditional Random Fields," in *Proceedings of Symposium in Information and Human Language Technology*, Natal, RN, Brazil, Nov. 2015, pp. 27–31.

[9] Y. Benajiba, P. Rosso, and J. M. BenedíRuiz, "ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy," in *Computational Linguistics and Intelligent Text Processing*, vol. 4394, A. Gelbukh, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 143–153.

[10] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, Jun. 2001, pp. 282–289.

[11] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based Model for Arabic Language Understanding." arXiv, Mar. 07, 2021, https://doi.org/10.48550/arXiv.2003.00104.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv, May 24, 2019, https://doi.org/10.48550/arXiv.1810.04805.