IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 July 2023, accepted 4 October 2023, date of publication 12 October 2023, date of current version 27 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3324288

RESEARCH ARTICLE

Preserving Privacy in Arabic Judgments: AI-Powered Anonymization for Enhanced Legal Data Privacy

TAOUFIQ EL MOUSSAOUI, LOQMAN CHAKIR, AND JAOUAD BOUMHIDI, (Member, IEEE)
LISAC Laboratory, Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez 30003, Morocco

Corresponding author: Taoufiq El Moussaoui (taoufiq.elmoussaoui@usmba.ac.ma)

This work was supported in part by the Ministry of Higher Education, in part by the Scientific Research and Innovation, in part by the Digital Development Agency (DDA), and in part by the Moroccan National Center for Scientific and Technical Research under Grant Alkhawarizmi/2020/36.

ABSTRACT Jurisprudence involves studying, interpreting, and applying the law to comprehend its societal impact. Judges annually review cases to ensure accurate law application, which raises privacy concerns when accessing files from other courts. While the legal field has garnered interest from the research community, the challenge of masking personal data, particularly in the Arabic language with limited resources, remains in its early stages. To address this research gap, we develop a two-component system for generating anonymous Arabic judgments. The first component, a personal data extractor model, utilizes Named Entity Recognition (NER) to identify key individual entities like names, addresses, birthdays, case numbers, and national identity codes. We train this model on a purpose-built Arabic legal corpus. The second component involves a Python module designed to mask the personal entities extracted by the first component. Together, these components enable the generation of anonymous judgments. Our model achieves an F1-score of 96.14% when detecting entities in the created Arabic Legal corpus. Additionally, experiments on the ANERCorp corpus, with training and testing splits of 70%-30% and 90%-10%, yield F1-scores of 93.78% and 95.77%, respectively. With these results, our proposed system demonstrates the promising potential for generating anonymous Arabic judgments. Furthermore, the built Arabic legal corpus provides a valuable resource for researchers aiming to enhance domain-specific NER models in Arabic text.

INDEX TERMS Anonymous judgments, jurisprudence, personal data, Arabic named entity recognition, ANERCorp.

I. INTRODUCTION

Jurisprudence refers to the study, interpretation, and application of the law. It encompasses the philosophy, principles, and theories underlying the legal system. Jurisprudence seeks to understand the nature of law, its origins, its purpose, and its effects on society. It involves analyzing legal concepts, doctrines, and legal reasoning to develop a deeper understanding of the law and its implications. Jurisprudence also involves examining the role of judges, legal institutions, and the relationship between law and morality. It provides a framework for legal scholars, practitioners, and policymakers to analyze and debate legal issues and shape the development of the legal system.

The process of jurisprudence involves judges examining previous court cases and decisions to establish legal principles, interpret laws, and ensure consistency in their application. In doing so, judges may access files and information from other courts to gather insights and precedents that aid their decision-making process. While this practice is crucial for maintaining legal coherence and fairness, it raises privacy concerns. There is always a risk of inadvertent breaches or unauthorized access, which can compromise people's privacy.

By implementing anonymization techniques, jurisprudence seeks to strike a delicate balance between the need

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna Dulizia.

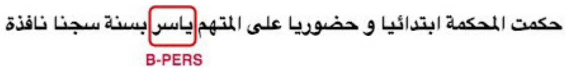

FIGURE 1. Example of a sentence that contains a named entity.

for access to data and the imperative to protect the privacy and personal information of individuals involved in legal matters. Anonymous judgments provide valuable resources for jurisprudence. They allow in-depth analysis and study of legal principles, reasoning, and outcomes without revealing sensitive personal information. Researchers can examine patterns, trends, and legal interpretations across a broad range of cases, thereby advancing the understanding of legal principles and their application.

By making anonymous judgments publicly available, courts promote the idea of open data. Open data [1] initiatives aim to increase access to information for the public, legal professionals, and researchers. This accessibility facilitates transparency, accountability, and the dissemination of legal knowledge. It empowers individuals to understand and engage with the legal system while fostering a more inclusive and participatory legal environment.

Text mining [2] is an approach for studying large volumes of textual data, intending to extract unknown relations and propose solutions to aid decision-making. Various technologies are utilized in the text mining process, including legal document summarization, translation, categorization, and information extraction. Named Entity Recognition (NER) [3] is an adopted technique in the extraction of entities from judgments, where it involves identifying and classifying named entities in text, particularly relevant nouns such as persons (PERS), locations (LOC), and organizations (ORG). Figure 1 presents an example of a sentence that contains a named entity.

The majority of the used systems primarily handle English and European languages. However, there has been comparatively less emphasis on named entity extraction from Arabic text. Furthermore, the few existing Arabic Named Entity Recognition (NER) systems typically prioritize general domain entities like names, addresses, and numbers rather than specific legal domain entities such as the case number and the national identity code. The lack of resources in Arabic, particularly legal entity corpora contributes to this focus on general entities.

We have developed a two-component system to address the challenge of generating anonymous judgments. The first component, a personal data extractor model, employs the NER concept to extract entities that identify individuals. We trained this model on a purpose-built Arabic legal corpus. The second component involves a Python module designed to mask the personal entities extracted by the first component. Together, these components ensure the generation of anonymous judgments, safeguarding the privacy of the individuals involved.

Overall, we summarize our contributions as follows:

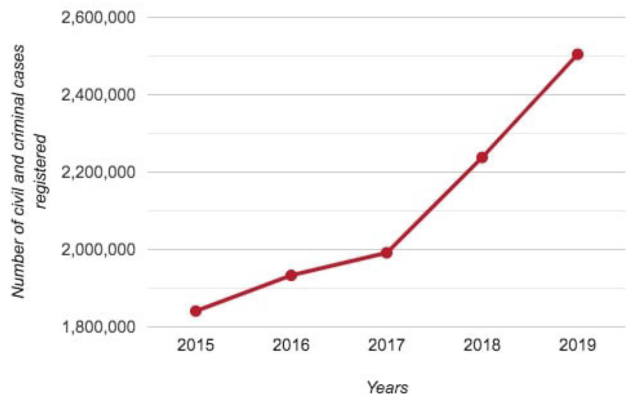

FIGURE 2. Evolution of the number of civil and criminal cases registered in Moroccan courts.

- We have developed an Arabic system that generates anonymous judgments based on original judgments.
- We have constructed a NER Arabic legal corpus to support our research.
- We demonstrate the practicality of our system by applying it to real judgments issued by the Moroccan courts.

The rest of the paper is structured as follows: section II explains the problem. Section III presents the related literature. In section IV, we give details on the corpus created, and the anonymous judgment generation system. Section V details the experimental setup, evaluation measures, results, and an ablation study. In Section VI, we provide the limitation and the directions for future work. Finally, conclusions are drawn in Section VII.

II. BACKGROUND

In this section, we provide a concise overview of jurisprudence in Morocco. We outline the existing approach for hiding personal data in judgments and introduce our proposed system.

Jurisprudence has gained significant importance in Morocco's legal field due to the exponential growth of daily legal verdicts issued by Moroccan courts. Figure 2 shows the evolution of the number of civil and criminal cases registered.

In 2006, two lawyers launched the first free portal for Moroccan case law decisions. In 2015, the National Agency for Telecommunications Regulation and the Court of Cassation signed an agreement for a project that will allow the dematerialization of the court's judgments, as well as the decentralization of their delivery to the interested parties at the level of all substantive courts.

In 2018, Morocco launched **Jurisprudence.ma** which is a platform that shares the latest case law in Morocco and Moroccan legal science. It also brings legal professionals closer to legal information.

Nowadays, the process of hiding personal data is manual. A court clerk is occupied by reading the judgments so he can identify the personal data to mask it by replacing it with

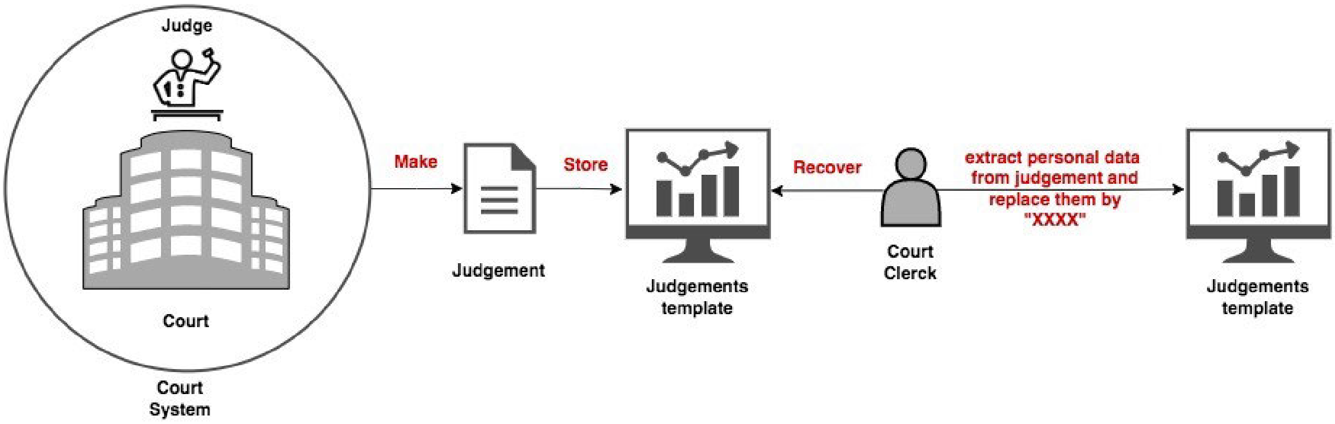

FIGURE 3. The current process for hiding personal data in judgments.

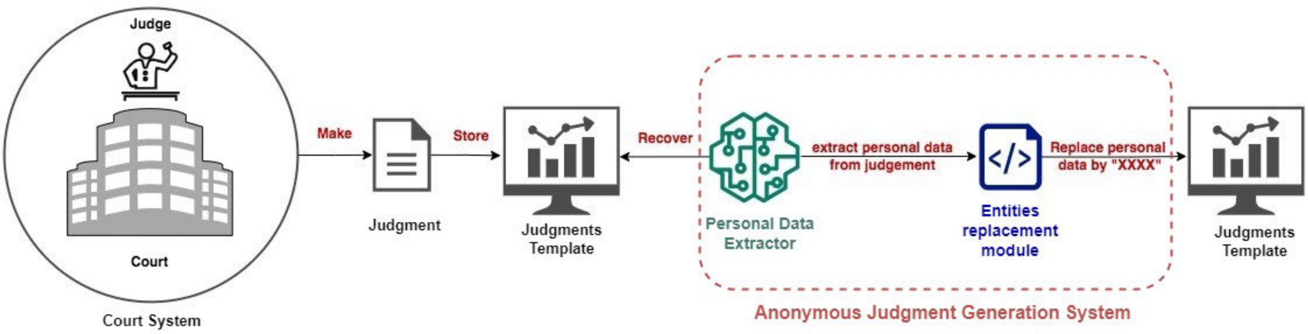

FIGURE 4. New process for hiding personal data with the proposed system.

the character "XXXX". According to the CNDP, which is the Moroccan Commission for the Control of personal data protection, and the European Commission, personal data are all information relating to an identified individual. A person can be identified:

- **Directly:** by name or surname.
- **Indirectly:** by national identity code, birth date, social security number, a postal or e-mail address.

Figure 3 illustrates the current process of generating anonymous judgments. After a judge announces a judgment, it is recorded in a dedicated legal platform. Subsequently, a court clerk retrieves the judgment, carefully examines its content, and identifies any personal data (such as names, locations, national identity codes, birthdates, etc.) associated with the mentioned parties. To protect privacy, the court clerk replaces this sensitive information with the character "XXXX". However, this manual process has several disadvantages. It is time-consuming, prone to errors during data extraction, and fails to respect citizens' privacy. In response to these drawbacks, we have developed a system that offers the same service in a shorter time. Additionally, our system provides open data, empowering citizens to participate in the development of traditional policies. By increasing government transparency through open data, citizens can access information behind reports and contribute their opinions. Moreover, it creates opportunities for collaboration between the public and private sectors.

Our work involves the application of named entity recognition (NER) to develop a personal data extractor. Additionally, we have created a Python module that automatically replaces the extracted data with "XXXX," ensuring the usability of the anonymous judgment for judges. Figure 4 illustrates the new process implemented through our system, which replaces the manual involvement of the court clerk. The system comprises two components, offering an improved and streamlined approach.

III. RELATED WORK

This section reviews previous research on the use of AI approaches to legal documents. Then, we discuss earlier studies on the approach employed in our system (Arabic-named entity recognition), also the corpora used for this task, and we conclude with a synthesis that highlights our paper's contribution.

A. APPLICATIONS OF ARTIFICIAL INTELLIGENCE TECHNIQUES ON LEGAL DOCUMENTS (JURISPRUDENCE)

Recently, researchers have attempted to analyze legal data and utilize the vast volumes of data generated by the courts to aid in decision-making. The work discussed in reference [4] uses deep learning approaches to address the issue of automatically categorizing Islamic law concerns. They create a hierarchical deep learning model that extracts the word and sentence characteristics from the question text first, then applies a text classifier to the question representation. The research described in reference [5] tries to provide a response to the following question: Given a case and its set of facts, what are the cases that are closely linked to it and may be utilized as sources to develop arguments for the provided case? By implementing CNN, LSTM, and Doc2Vec models for multi-class multi-label text classification, they investigate the question of locating relevant court decisions from a recent legal case. Research [6] builds a Word Embeddings model from Philippine Jurisprudence. They use a sizable corpus of judgments, resolutions, and opinions from the Philippine Supreme Court from 1901 to 2020 to train nine-word embedding models. They measured their performance in terms of accuracy on a specially created word analogies exam consisting of 4,510 questions divided into seven syntactic and semantic categories. They found that FastText models performed better on syntactic evaluators than Word2vec models did on semantic evaluators.

An interesting study [7] described a legal assistant powered by AI, which would facilitate case evaluation using justifications taken directly from judicial decisions. This system gives legal professionals a useful tool to assist their clients in a lawsuit, without needing to review every case law. The work discussed in reference [8] aims to use NLP to extract legal parties from text documents containing legal opinions. The paper suggests a deep-learning approach that may successfully be applied to address the issue of detecting legal party members in legal documents. The authors propose a BRNN model to determine if an entity is a legal party or not, and a modified version of the same neural network to categorize the discovered entities into petitioner and defendant classes.

The authors of the research [9] provide an approach to prioritize the pertinent legal principle citations in court cases that support a certain motion. The system generates two scores: one is based on word embedding text similarity, whereas the other is based on feature significance metrics, where each legal article is a feature fed to a classifier for decision.

The study [10] presents the role of AI in patent law debates. The authors discuss the use of AI to help patent offices and applicants assess the patentability of their inventions by analyzing massive amounts of documents in a shorter time frame. They also work on the use of AI to perform patent essentiality checks.

B. ARABIC NAMED ENTITY RECOGNITION (ANER)

Multiple features may be retrieved from the immense amount of data row available on social media. Reference [11] presents a study that aims to identify Arabic-named entities on social media using two embeddings and a multi-headed self-attention mechanism applied in the BiLSTM-CRF structure. In their method, character and word embedding are combined at the embedding layer, and the attention mechanism computes the similarity over the sequence of characters and records local context data. The experiments demonstrate that their method provides satisfactory results for social media datasets.

Ali et al. [12] created a NER model based on BiLSTM and CRF. They utilize Madamira to provide morphological and syntactical properties to various word embeddings. According to the experiments, the extra features help the model perform better.

The study [13] proposes a CRF-based approach with part-of-speech tags. They created an Urdu NER dataset (UNER-I) that was manually annotated with a significant number of NE categories. Experiments were carried out utilizing the created dataset and an existing dataset to measure the performance of the suggested technique and the usability of the dataset. The results showed that the proposed technique reaches good scores.

The authors of [14] suggest a Bi-LSTM model for this task. This network is suited for the NER task since it can examine sequences and link each component. Furthermore, they employed pre-trained word embedding to train the inputs to the Bi-LSTM network. The suggested model is tested using the ANERCorp corpus, and according to the experimental data, the word embedding model achieves a high F1 score of 87.12%.

The authors of [15] describe and evaluate a character marker for Arabic NER. A sentence is presented as a list of characters rather than words. The model consists of stacked two-way LSTMs that take characters and produce labeling probabilities for each character. These probabilities are transformed into word-level consistent named entity labels using a Viterbi decoder [16].

Jarrar et al. [17] created a corpus of nested named entities in Arabic. They used it to create a nested NER model based on the pre-trained AraBERT. Overall, the model scored 88.40% on the F1 score.

Mousa et al. [18] proposed a hybrid model combining RBF, CNN, and BiLSTM for Arabic-named entity recognition (NER). Using the ANERCorp dataset with a 90% training and 10% testing split, they achieved an impressive 95% f-score. The hybrid model surpasses stand-alone models (MLP, KNN, NB, SVM, RBF, CNN, BiLSTM), showcasing its superior performance in Arabic NER.

To generate the word representation vectors, Affi and Latiri [19] employed variant deep neural network architecture combined with CNN, LSTM, and BERT. The CNN-LSTM-BERT Combinatorial feature embeddings with BiLSTM-CRF and CNN-LSTM-BERT Combinatorial feature embeddings with BiGRU-CRF achieved an F1-score of 93.34% and 93.68 % respectively.

C. ARABIC NAMED ENTITY RECOGNITION CORPUS

Numerous corpora have been developed specifically for the task of named entity recognition (NER) in Arabic. However,

TABLE 1. List of annotated datasets for Arabic NER. #Entities refers to the number of entity types.

Corpus	Year	Source	#Entities	URL
ACE 2003	2004	Transcribed speech, Newswire, Broadcast news	7	https://catalog.ldc.upenn.edu/LDC2004T09
ACE 2004	2004	Broadcast news, Newswire, Telephone conversations	7	https://catalog.ldc.upenn.edu/LDC2005T09
ACE 2005	2005	Broadcast conversation, Broadcast news, Newsgroups, Telephone conversations, Weblogs	7	https://catalog.ldc.upenn.edu/LDC2006T06
ACE 2007	2007	Weblogs, Newswire	7	https://catalog.ldc.upenn.edu/LDC2014T18
ANERcorp	2007	Newswire and Web	4	https://camel.abudhabi.nyu.edu/anercorp/
REFLEX	2009	Newswire	4	https://catalog.ldc.upenn.edu/LDC2009T11
AQMAR	2012	Arabic Wikipedia articles	7	http://www.cs.cmu.edu/ ark/ArabicNER/
OntoNotes 5.0	2013	Telephone conversations, Newswire, Newsgroups, Broadcast news, Broadcast conversation, Weblogs, Religious texts	18	https://catalog.ldc.upenn.edu/LDC2013T19
WDC	2014	Wikipedia	4	https://github.com/Maha-J-Althobaiti /Arabic_NER_Wiki-Corpus
DAWT	2017	Wikipedia	4	https://github.com/klout/opendata/tree/ master/wiki_annotation
CANERCorpus	2018	Religious texts (Hadiths)	14	https://github.com/RamziSalah/Classical-Arabic -Named-Entity-Recognition-Corpus
Wojood	2022	Wikipedia	21	https://sina.birzeit.edu/wojood/

it is important to note that these corpora primarily cover the general domain, with annotations focusing on entities of a general nature. The identified entity types within these corpora include PERSON, DATE, TIME, LOCATION, ORGANIZATION, MONEY, PERCENT, NATIONALITY, and RELIGION. Table 1 summarizes the corpora used for the Arabic NER task [20].

D. SYNTHESIS

Overall, the aforementioned studies have highlighted important works done in the legal domain, also in recognizing Arabic-named entities. However, the application of NER to legal corpora is still in its early stages, and the integration of AI in the legal field has yet to effectively tackle the issue of masking personal data. Addressing this challenge would significantly enhance the use of AI in the legal field, enabling more secure and privacy-conscious applications for legal professionals, researchers, and organizations dealing with legal corpora. Additionally, existing Arabic NER corpora predominantly concentrate on general domains, lacking specific legal corpora. While these existing corpora provide valuable resources for Arabic NER research, their applicability to domain-specific contexts, such as law, medicine, or finance, remains limited. Thus, there is a compelling need to construct domain-specific Arabic NER corpora to address the unique challenges and requirements within these specific domains. The development of such corpora would enable researchers to enhance the precision and contextual relevance of NER models tailored to the domain-specific Arabic text, facilitating more accurate and comprehensive named entity recognition in specialized fields.

IV. RESEARCH METHODOLOGY

In this section, we outline the creation process of our Arabic legal corpus, which serves as the foundation for training and evaluating our system. Next, we introduce our system designed to generate anonymous judgments, providing an overview of its architecture and the details of each component.

A. ARABIC LEGAL CORPUS

The purpose of creating an Arabic Legal Corpus is to provide training data for our system's model so that it can automatically recognize and extract the legal named entities from unannotated text.

1) DATA COLLECTION PROCESS

To build the corpus, we rely on the judgments, which are the documents generated after the judicial hearing. Each one contains information about a trial. Figure 5 presents its structure and an example of each section of the judgment document. Each judgment comprises five parts, the first of which contains information about the court's counsel, more specifically, the names of the chief of the court, the clerk, and the representative of the public prosecutor's office, we also found the case number and the court address. The second part contains information about the victim and the culprit, such as their names, dates of birth, addresses, and national identity codes. The third part tells the story of what happened between the victim and the culprit. The fourth part presents the laws on which the judge based his decision, and the final section is the judge's decision.

The Moroccan Commission for the Control of personal data protection (CNDP), and the European Commission defines personal data in a judgment as names, addresses, birth dates, national identity codes, case number, vehicle plate number, etc. Those informations could be found in the whole document except the fourth part which contains only the laws used for judgment.

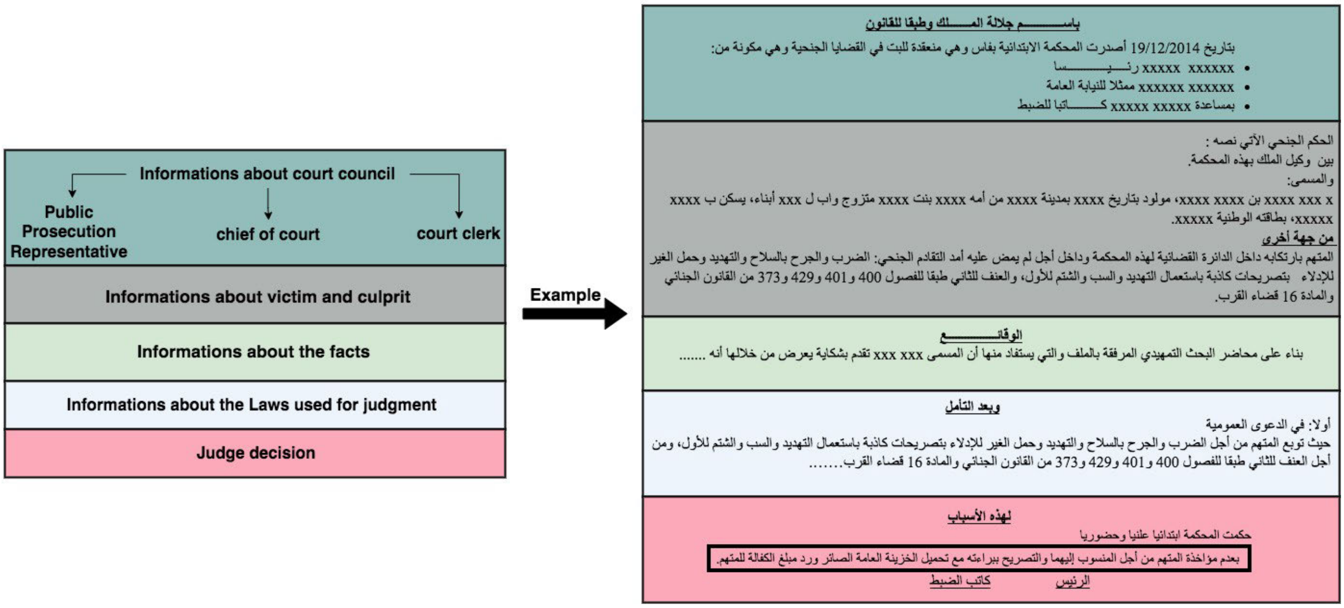

FIGURE 5. Structure of the judgment with an example of each part of it.

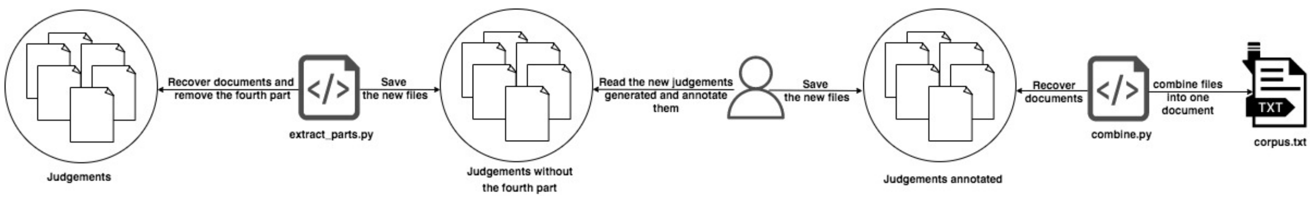

FIGURE 6. The Corpus creation process.

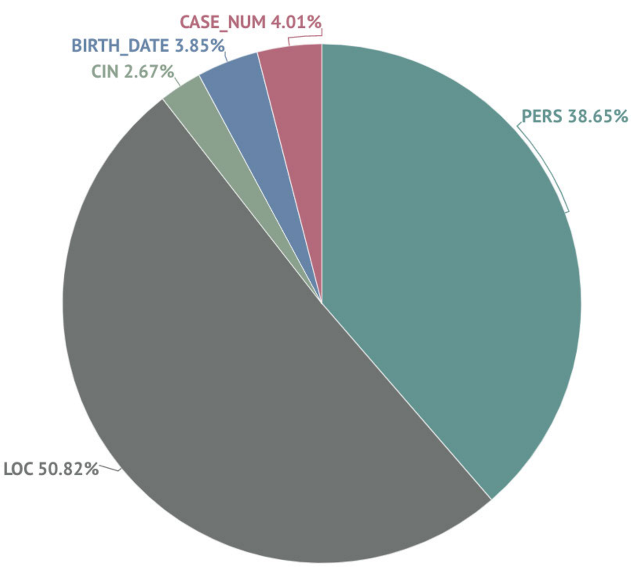

FIGURE 7. Distribution of named entities in the corpus created.

In the creation process of the Arabic legal corpus, we will not use the entire document but only the parts that contain the personal data. For that, we develop a Python script named: *extract_parts.py* that takes as input the whole judgment document and generates a file without the fourth part. The saved text is divided by term, each term is in one line. Then we manually labeled each file generated using IOB tagging following the annotation strategy described in the MUC-6 [21]. For the annotation, the following words were used: B-PERS, I-PERS, B-LOC, I-LOC, B-CIN, B-BIRTH_DATE, B-CASE_NUM, and O. Where PERS means a named entity of a person, LOC signifies a named entity of an address, CIN denotes a named entity of the national identity code, BIRTH_DATE signifies a named entity of a birthday date, CASE_NUM means a named entity of the case number, and O denotes other words that are not named entities. The last step in the process of creating the corpus is combining all the labeled txt files into a single file (corpus.txt), for this we have developed another Python script named: *combine.py* that does this and adds a ''. O'' line as a separator between txt files. Figure 6 presents the process for corpus creation.

2) DATA EXPLORATION

The corpus created (corpus.txt) contains 682458 tokens and 114781 named entities from 4795 judgments. Entities were

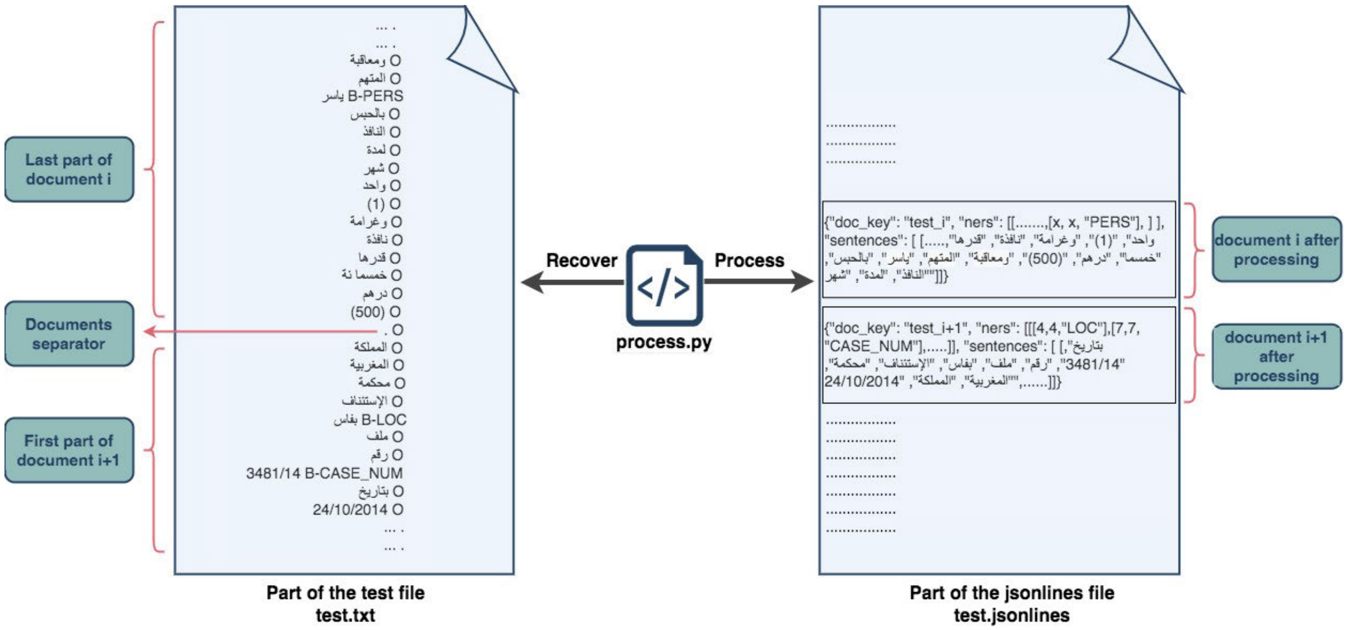

FIGURE 8. Example of the processing process.

TABLE 2. Statistics of the entities in the created corpus.

Entities	Train	Test	Corpus
PERS	34162	10205	44367
LOC	44340	14003	58343
CIN	2514	553	3067
BIRTH_DATE	3492	929	4421
CASE_NUM	3528	1055	4583
O	488153	79524	567677
Total	576189	106269	682458

distributed as 38.65%, 50.82%, 2.67%, 3.85%, and 4.01% for person, location, national identify code, birthday date, and case number respectively. Figure 7 presents the distribution of named entities in the corpus created.

The corpus is divided into two parts: One presents 80% of the total corpus used for training the model, and it contains 576189 tokens and 88036 named entities. The remaining 25% contains 106269 tokens and 26745 named entities. This part is used for the performance test. Table 2 presents the statistics of the entities in the created corpus.

3) DATA PROCESSING

We apply the preprocessing process on both files train.txt and test.txt. The process consists of transforming each one into jsonlines form. Each file contains multiple documents that are separated by ". O". Each document is divided into two columns, the first presents the words, and the second is for tags (named entities).

The preprocessing consists to transform the file with the two columns to a jsonlines file that contains a list of dictionaries. Each document of the file is converted to a dictionary with three keys. The first key is named **"doc_key"** which is the concatenation of the type of the file (train or test) with the index of the document in the file, this attribute presents the key to each dictionary. For example, the value of the tag doc_key of the first dictionary of the test file is test_0. The second key is **"ners"**, which is a list of sub-lists when each one has the following structure: [i, j, tag] where i presents the position of the beginning of the tag, j is the position of the end of the same tag and tag is the named entity type. The last key is **"sentences"**, which is a list of sub-lists when each sub-list contains the terms of the sentence. We developed a Python script (*process.py*) that does this processing. Figure 8 presents an example of the processing process.

B. ANONYMOUS JUDGMENT GENERATION SYSTEM

The introduced anonymous judgment generation system focuses on producing anonymized judgments derived from the original ones. The system comprises two components: a personal data extractor model for identifying legal named entities, and a replacement module that replaces the entities recognized with the character "XXXX".

1) PERSONAL DATA EXTRACTOR MODEL

In this work, we follow the common architecture of deep learning-based NER models, which comprises three key layers: the input layer, the encoder layer, and the prediction layer. We concatenate word and character embedding and feed it to BiLSTM to get the input representation layer. We use two separate feed-forward neural networks as a context encoder layer and a biaffine classifier as a prediction layer. Figure 9 illustrates the architecture of our model.

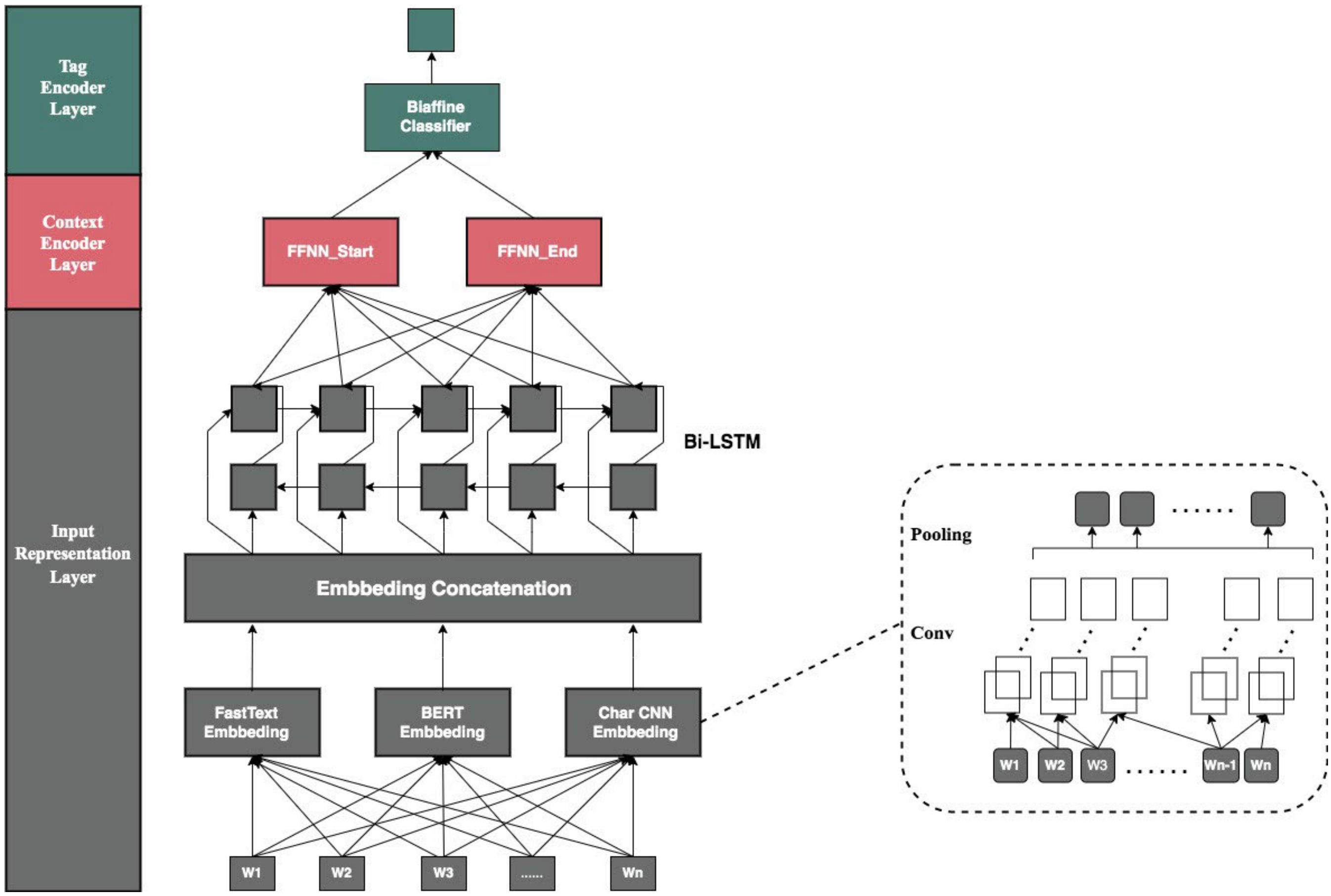

FIGURE 9. The architecture of the personal data extractor model.

2) INPUT REPRESENTATION LAYER

Distributed representation involves representing words as dense, low-dimensional real-valued vectors, with each dimension representing a latent feature. It captures both semantic and syntactic properties of words, which may not be explicitly present in the input. In our case, we use a hybrid representation of the text, where we concatenate the word-level and the character-level representation.

For the word-level representation, we use both the Arabic BERT and FastText. FastText [22] is a toolkit developed by the Facebook Research Team, designed to effectively learn word representations. One of the main advantages of the fastText embedding model is its consideration of the internal structure of words during the learning process. This aspect proves particularly beneficial for morphologically rich languages like Arabic. Unlike other word embeddings, such as word2vec, fastText adopts a different word representation approach. It views a word as a composition of n-grams of characters, with the length of n varying from 1 to the word's length. The benefit of this approach lies in its ability to find vector representations for words that might not be directly present in the dictionary.

BERT is a bidirectional contextual language model based on the Transformer architecture. To obtain BERT embeddings for sequences of unlimited length, we propose to use the more recent BERT embeddings [23].

Let W be a fixed window length, we obtain a representation for token t by applying BERT to the sequence of tokens from W to the left of t to W to the right of t. We then take the five last layers of the BERT model for a token t and apply a learnable weighted averaging. The output of the network is taken as the representation of token t. We use $W = 64$, since using the maximum size of $W = 256$ is computationally intensive, and good results are already obtained with 64.

For the character-based word embeddings, we use a CNN to encode the characters of the tokens. Each word in the input sequence is segmented into its characters. As shown in Figure 10, individual characters are fed into an embedding layer with randomly initialized vectors of dimension d to obtain their representations.

To represent token t, which has a length of l, we transform it into a matrix M of size $d{*}l$. Additionally, we create a convolutional filter H of size $d{*}n$, where n is the filter width, and the values in H are randomly initialized. The convolutional operation between M and H is conducted, resulting

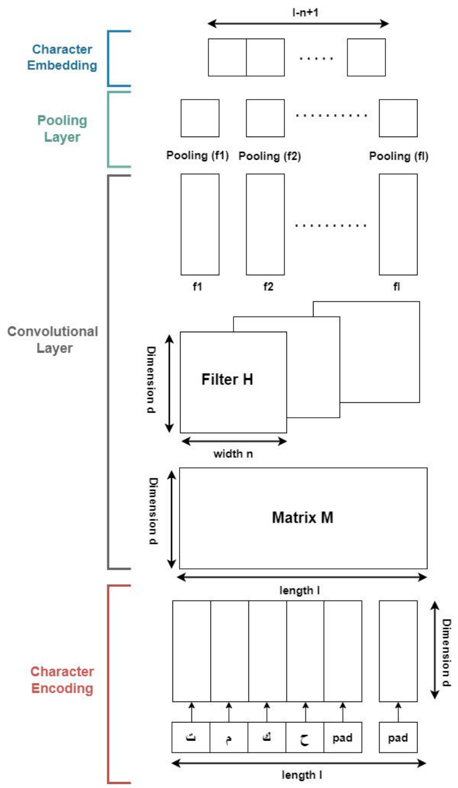

FIGURE 10. Model for character embedding based on Convolutional Neural Networks (CNN).

in a feature map f with a size of l-n+1, as demonstrated in Equation 1. The feature map is generated using the Frobenius inner product, denoted by $\langle M, H\rangle$.

$$f_k =< M[*, k : k+n-1], H > \tag{1}$$

Then, the pooling operation is applied to extract a single feature for all feature maps as Equation 2 shows. y_H^t is the character representation of a word t given by the filter H.

$$y_H^t = Pooling(f_k) \tag{2}$$

The process is iterated multiple times using various convolutional filters (4, 5, and 6), and their outputs are appended to create the final word representation.

After generating the Character-level Representation with CNN and the Word-level Representation with fasttext and Bert, we concatenate them as the example shown in the Figure 11. Then, we apply a BiLSTM recurrent network on the top of the embedding concatenation layer to get the representations (z) of the word.

The word t_k in an input sentence is encoded into a single vector from its concatenated embeddings using Equation 3:

$$\begin{aligned} z^k &= BILSTM(t_k) \\ &= BILSTM(Emb^k_{concat}) \end{aligned} \tag{3}$$

where the token t_k is presented by its concatenated embeddings Emb^k_{concat}. Figure 12 illustrates the process of getting

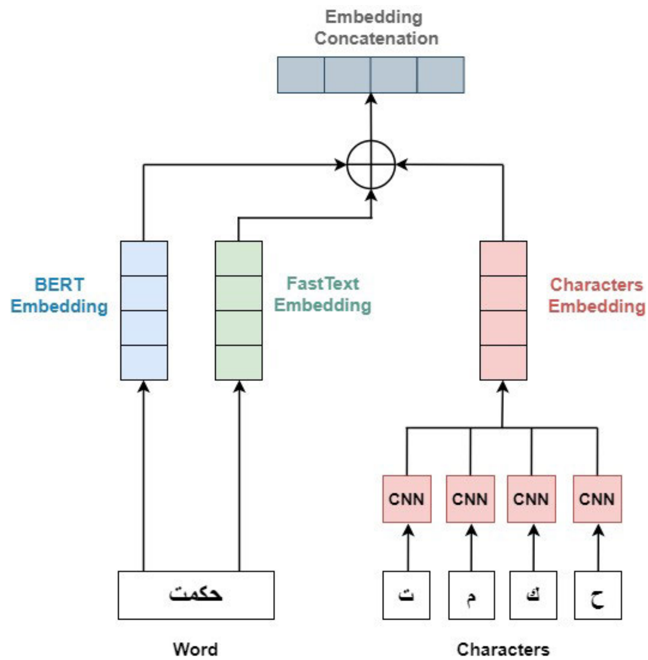

FIGURE 11. Example of concatenated embedding. We concatenate the outputs of character embedding from CNN and word embedding from BERT and FastText to obtain its final representation.

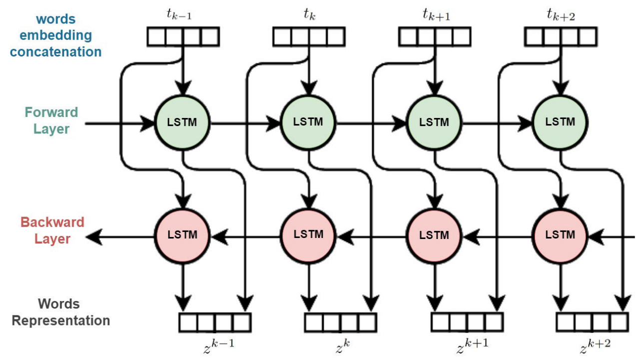

FIGURE 12. Application of BiLSTM over the embedding concatenation layer to get the final representation of the words.

the word representation by applying the BiLSTM over the embedding concatenation layer.

3) CONTEXT ENCODER LAYER

The context encoder layer of our model is presented by two feed-forward neural networks. The Feedforward neural network is a fundamental type of Artificial Neural Network (ANN) that operates in a forward direction. It consists of an input layer, multiple hidden layers, and a final output layer. Neurons within the network are interconnected by weights, representing the probability-weighted associations between input and output. During training, the network computes the differences between the processed output and the desired target output. It adjusts the weights of each connection based on a learning rate and the error values. This iterative process allows the network to progressively converge toward producing outputs that closely match the target results.

We apply two separate feed-forward neural networks (FFNN) to create various representations *(REP_b/REP_e)* of the spans (Equations 4 and 5). The advantage of having multiple representations of the beginning/end spans is that the system can learn more about how can recognize the beginning/end independently, which improves accuracy.

$$REP_b(x) = FFNN_b(z_{bx}) \tag{4}$$
$$REP_e(x) = FFNN_e(z_{ex}) \tag{5}$$

4) PREDICTION LAYER:

The prediction layer of our model utilizes the biaffine model. By applying this model to the sentence, we generate a scoring tensor ($scor_w$) using Equation 6. The scoring tensor provides scores for all possible spans representing named entities. These spans follow the constraint that the start of the entity $ne_b(x)$ is before its end $ne_e(x)$.

$$scor_w(x) = REP_b(x)^T \cup_w REP_e(x) + W_w(REP_b(x) \oplus REP_e(x)) + E_w \tag{6}$$

We calculate the score of each span and assign it to a named entity (ne) using Equation 7. To prioritize spans, we rank them in decreasing order based on their scores. However, we ensure that lower-scored spans do not interfere with the borders of higher-scored entities.

$$ne(x) = argmax(scor_w(x)) \tag{7}$$

The model has been optimized using softmax cross-entropy (Equations 8 and 9).

$$p_w(x_n) = \frac{\exp(scor_w(x_n))}{\sum_{n=1}^{N} \exp(scor_w(x_n))} \tag{8}$$
$$loss = -\sum_{x=1}^{L}\sum_{n=1}^{N} ne_{xn} \log p_w(x_n) \tag{9}$$

5) ENTITIES REPLACEMENT MODULE

The Entities Replacement Module is a Python module specially designed to hide personal entities in a judgment document. It is an integral part of the data anonymization process, with the objective being to protect the privacy of the individuals mentioned in the judgment.

The module needs the original judgment document as input. In addition, it relies on the results of the personal data extraction model, which provides a list of identified entities present in the judgment. These entities include sensitive information such as names, addresses, dates of birth, and other personally identifiable details.

Once the module receives the input, it systematically scans the entire judgment document, diligently searching for instances of the identified personal entities. Upon locating an entity, it performs the replacement operation, substituting the entity with the character 'XXXX'. The Output of the Entities Replacement Module is an anonymized judgment.

TABLE 3. Hyper-parameters of the personal data extractor model.

Layer	Parameter	Value
Input Representation	Arabic FastText embedding size	300
Input Representation	BERT size	1024
Input Representation	Embedding dropout	0.5
Input Representation	Character embedding size	9
Input Representation	CNN character size	50
Input Representation	Character CNN filter widths	4,5,6
Input Representation	BERT layer	Last 5
Input Representation	Size of Bi-LSTM	200
Input Representation	Bi-LSTM layer	3
Input Representation	Bi-LSTM dropout	0.4
Context Encoder	Size of FFNN	150
Context Encoder	FFNN dropout	0.2
Context Encoder	Optimizer	Adam
Context Encoder	Learning rate	1e-3

V. EXPERIMENTAL SETUP AND RESULTS

In this section, we present the experimental setup as well as the parameters utilized to train and test the model. After that, we detail the evaluation measures, then we highlight the results, and discuss them. Finally, we present an ablation study that seeks to assess the contributions of various system parts individually.

A. EXPERIMENTAL SETUP

We use the dataset created (See section IV-A) to evaluate our data extractor model. The model is configured by the parameters mentioned in table 3.

B. EVALUATION MEASURES

We evaluated the model's performance using multiple metrics. We started by computing **Precision** that quantifies the number of correct positive predictions made. The precision measure is calculated using Equation 10:

$$Precision = \frac{True\_Positives}{True\_Positives + False\_Positives} \tag{10}$$

The second metric is **Recall**, and it is calculated using Equation 11:

$$Recall = \frac{True\_Positives}{True\_Positives + False\_Negatives} \tag{11}$$

Third, **F1-score** which is a metric that can be calculated using precision and recall as Equation 12 shows:

$$F1 - score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{12}$$

C. RESULTS

To evaluate our model, we utilized various deep learning models, starting with the Bidirectional Long Short-Term Memory algorithm. We then enhanced it by combining a convolutional neural network (CNN) for character representation with Bi-LSTM. Finally, we incorporated CRF into the combination. The performance of our model, along with other deep learning-based models, is presented in Table 4, while the detailed evaluation findings are illustrated in Table 5.

TABLE 4. Performance of personal data extractor model against other deep learning models.

Models	Precision (%)	Recall (%)	F1-score (%)
Bi-LSTM	92.71	92.46	92.58
CNN-BiLSTM	94.06	93.52	93.78
CNN-BiLSTM-CRF	94.91	94.22	94.56
Our model	**96.43**	**95.86**	**96.14**

TABLE 5. Results of the personal data extractor model.

	Precision (%)	Recall (%)	F1-score (%)
PERS	96.94	96.36	96.64
LOC	95.24	94.51	94.87
CIN	98.37	97.84	98.10
BIRTH_DATE	97.03	96.11	96.56
CASE_NUM	97.54	96.91	97.22
Overall (%)	96.43	95.86	96.14

Furthermore, we compared the performance of our proposed model with several previous studies conducted on the ANERCorp dataset. Table 6 showcases the results of competitive Arabic NER models from the studies of Naji F. Mohammed (2012), David Awad (2018), Mohammed N. A. Ali (2018), Mourad Gridac (2018), Mohammed Nadher Abdo Ali (2019), Mohammed N. A. Ali (2019), Aya Mousa (2020), Ismail El Bazi (2020), Wissam Antoun (2020), Alsaaran Norah (2021), and Manel Affi (2022).

D. ANALYSIS AND DISCUSSION

Table 4 shows that our model outperformed other deep learning models in terms of the defined metrics. In term of F1-Measure, our proposed model outperforms the BiLSTM model by **3.56**, the CNN-BiLSTM model by **2.36** as well as the CNN-BiLSTM-CRF model by **1.58**. The processing done in the input representation layer is the key to this improvement. Concatenating the context semantic representation of the word vectors generated by the BERT model, character embeddings, and context-independent embeddings with Fast text yields a good representation of sentences; additionally, transferring the knowledge represented to BiLSTM aids in the learning of additional context information.

Table 5 shows the details of the model's evaluation findings. We notice that the **CIN** entity is the most recognized entity by the model since it has the highest values and the reason for that is that the values of this entity follow a specific pattern while the value of F1-measure of the **LOC** entity does not exceed 95% and this is because of the complexity of this entity which is generally composed unlike the other entities.

Table 6 presents the results of previous work on Arabic-named entity recognition using ANERCorp corpora. The table is divided into two parts based on the corpus splitting percentage. The models developed by the authors of [18] were tested on a test sample of 10%, and the training was done on 90% of the corpus. To compare the performance of our model with their models, we followed the same splitting strategy as they. Their best model is a hybrid model which consists of a Radial Basis Function (RBF) cascaded with a sequential Convolutional Neural Network (CNN) and Bidirectional Long-Short Term Memory (BiLSTM). As Table 6 shows, our model surpasses their model by **0.77** and **2.79** points in terms of f1 measure and recall respectively. Though our model slightly lags in precision, losing to their model by **1.57** points.

The majority of works on Arabic-named entity recognition using ANERCorp used the default split provided by the corpus owners, i.e. 70% for training and 30% for testing. The second part of Table 6 presents a sample of the best works, which proposed models with higher scores. We perform an experiment using the same corpus as them to compare the performance of our model. As shown in Table 6, the top 3 architectural models that achieve an f1 measure of over 93% are CNN-LSTM-BERT Combinatorial feature embeddings + BiGRU-CRF, CNN-LSTM-BERT Combinatorial feature embeddings + BiLSTM-CRF and our model. Our model surpasses CNN-LSTM-BERT Combinatorial feature embeddings + BiGRU-CRF by **0.1**, **0.05**, and **0.14** in terms of f1 measure, recall, and precision respectively. Also, it surpasses CNN-LSTM-BERT Combinatorial feature embeddings + BiLSTM-CRF by **0.44**, **0.32**, and **0.54** in terms of f1 measure, recall, and precision respectively.

The common point between the two models proposed by Affi and Latiri [19] in 2022 and our model is that the input representation layer is composed by concatenating multiple embeddings. The fact that our model uses these three types of embeddings together allows the model to capture both important morphological and orthographic patterns as well as semantic and syntactic word relationships.

Figure 13 shows an example of the application of our system on a judgment. The judgment on the left is the original judgment, and the one on the right is the anonymous judgement. As can be seen, the entities referring to names, addresses, case number, date of birth and national identity code have been well recognized by the system, and they have been hidden and replaced by "XXXX".

E. ABLATION STUDY

To measure the contribution of each component of the model, we run several experiments, and at each iteration, we removed a component and use the corpus created for the calculation of metrics. This ablation study aims to understand the importance of individual components.

1) CONTEXTUAL EMBEDDINGS

After removing the BERT embeddings component, the system performance drops significantly by 1.9 percent as Table 7 shows. This demonstrates that one of the most crucial elements for accuracy is BERT embeddings.

2) CONTEXT INDEPENDENT EMBEDDINGS

After eliminating the fastText embedding component from the system, we noticed that the score dropped by 0.4% (Table 7). This implies that the context-independent

TABLE 6. Comparison between the proposed model and other work on the ANERCorp dataset divided into 90%-10% and 70%-30% for training-testing respectively.

Model	Precision (%)	Recall (%)	F1 Measure (%)
Dataset Division : 90% for training and 10% for testing			
Multilayer Perceptron (MLP) [18]	60.33	42.67	44.67
K-Nearest Neighbors (KNN) [18]	86.67	77.33	81.33
Naïve Bayes (NB) [18]	29.00	33.33	29.67
Support Vector Machine (SVM) [18]	88.67	62.33	69.67
Radial Basis Function (RBF) [18]	88.33	78.33	83.00
Word2Vec + CNN [18]	85.33	78.33	81.67
Word2Vec + BiLSTM [18]	86.67	77.67	81.67
RBF + Word2Vec + CNN + BiLSTM [18]	**98.00**	92.33	95.00
Our Model	96.43	**95.12**	**95.77**
Dataset Division : 70% for training and 30% for testing			
CNN Char embeddings + AraVec 2.0 Word embedding -BiLSTM-Softmax [14]	88.49	87.57	88.01
CNN Char embeddings + AraVec 2.0 Word embedding -BGRU-Softmax [14]	87.73	86.51	87.12
CNN Char embeddings + BiLSTM-CRF [19]	90.62	90.43	90.52
LSTM Char embeddings + BiLSTM-CRF [19]	90.50	90.30	90.39
BERT Word embeddings + BiLSTM-CRF [19]	92.33	92.90	92.61
CNN-LSTM-BERT Combinatorial feature embeddings + BiLSTM-CRF [19]	93.20	93.50	93.34
CNN Char embeddings + BiGRU-CRF [19]	90.97	90.75	90.85
LSTM Char embeddings + BiGRU-CRF [19]	90.80	90.57	90.68
BERT Word embeddings + BiGRU-CRF [19]	92.58	93.10	92.83
CNN-LSTM-BERT Combinatorial feature embeddings + BiGRU-CRF [19]	93.60	93.77	93.68
BERT Word embeddings + BiGRU [24]	90.40	90.66	90.51
Multi-Attention [25]	90.43	90.62	90.52
CNN Char embeddings + Word embedding -BiLSTM-CRF [26]	-	-	90.60
AraBERT v0.1 [27]	-	-	84.20
AraBERT v1 [27]	-	-	81.90
Artificial neural network [28]	65.03	62.33	57.47
B-GRU Char embeddings + Word embedding -B-GRU-CRF [29]	91.00	88.52	89.74
CNN Char embeddings + Word2Vec Word embedding -BiLSTM-CRF [30]	84.15	68.74	75.68
Char embeddings + Word embedding -BiLSTM-BiLSTM Decoder [31]	93.52	90.54	92.01
Our Model	**93.74**	**93.82**	**93.78**

TABLE 7. The comparison between our full model and ablated models.

Models	F(%)	Delta (%)
Our model	96.14	
Our model WITHOUT BERT emb	94.24	1.9
Our model WITHOUT FastText emb	95.74	0.4
Our model WITHOUT Char emb	95.49	0.65
Our model WITH CRF Instead of Biaffine classifier	95.44	0.7

embeddings can still significantly enhance the system even with the BERT embeddings enabled.

3) CHARACTER EMBEDDINGS

Table 7 shows that the impact of character embeddings is significant, which demonstrates the benefit of adding character embeddings.

4) BIAFFINE CLASSIFIER

Instead of ablating the biaffine classifier, we replace it with the CRF, since it is frequently used in NER models. This modification transforms the system into a sequence labeling model. The experiment shows that the performance dropped by 0.7 percent, which indicates that the biaffine classifier is the best choice for the system.

Overall, the biaffine mapping and the concatenation of BERT embedding, and character embeddings present the best-performing system.

VI. LIMITATION AND FUTURE WORK

One limitation of our work is that the entities that our personal data extractor model can recognize are limited to PERS, LOC, CIN, BIRTH_DATE, and CASE_NUM, and this is due to the dataset of judgment that we use to create the Arabic legal corpus. 98% of those judgments concern civil and criminal cases, and only 2% concern cases related to traffic cases. This type of judgment contains an important entity that refers to personal data which is the vehicle's plate number. Our future work will focus on two stages. Firstly, we will collect a larger volume of traffic regulations judgments to update the Arabic legal corpus and re-train our personal data extraction model accordingly. This effort aims

FIGURE 13. Example of application of the system on a judgment.

to incorporate the vehicle's plate number as a recognized entity. Secondly, we aim to optimize the search operation for judgments, addressing a common challenge faced by court staff. In Morocco, where three million cases are processed annually, the current search process is hindered by the relational database storing the case documents, resulting in sluggish retrieval. To enhance the search operation, we propose developing a search engine that relies on indexed judgments, which contain only essential information necessary for individualizing the original judgment. Our system for generating indexed judgments will leverage BERT keyword extraction and our enhanced personal data model.

VII. CONCLUSION

Jurisprudence is a field of great interest to the academic community, given the vast number of legal documents generated annually. With the advent of modern technology, significant advancements in this field have become feasible. In this paper, we present our system that preserves the confidentiality of persons in legal documents. The model of the system can recognize five entities that identify a person. In terms of metrics, the model performed well and achieved good results by getting F-measure, precision, and recall values of 96.14%, 96.43%, and 95.86% respectively.

REFERENCES

[1] Y. Wang, A. Hassan, F. Liu, Y. Guan, and Z. Zhang, "Secure string pattern query for open data initiative," *J. Inf. Secur. Appl.*, vol. 47, pp. 335–352, Aug. 2019.

[2] C. Blake, "Text mining," *Annu. Rev. Inf. Sci. Technol.*, vol. 45, no. 1, pp. 121–155, 2011.

[3] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 50–70, Jan. 2022.

[4] W. H. AlSabban, S. S. Alotaibi, A. T. Farag, O. E. Rakha, A. A. A. Sallab, and M. Alotaibi, "Automatic categorization of Islamic jurisprudential legal questions using hierarchical deep learning text classifier," *Int. J. Comput. Sci. Netw. Secur.*, vol. 21, no. 9, pp. 281–291, 2021.

[5] M. A. Martija, J. Domoguen, and P. Naval, "How deep is your law? Predicting associations between cases in Philippine jurisprudence," in *Proc. TENCON IEEE Region 10 Conf. (TENCON)*, Oct. 2019, pp. 886–891.

[6] E. Peramo, C. Cheng, and M. Cordel, "Juris2vec: Building word embeddings from Philippine jurisprudence," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Apr. 2021, pp. 121–125.

[7] R. Rietveld, J. Rossi, and E. Kanoulas, "Distilling jurisprudence through argument mining for case assessment," in *Proc. 1st Int. Workshop AI Intell. Assistance Legal Professionals Digit. Workplace (LegalAIIA)*, 2019, pp. 19–22.

[8] C. Samarawickrama, M. de Almeida, N. de Silva, G. Ratnayaka, and A. S. Perera, "Legal party extraction from legal opinion texts using recurrent deep neural networks," *J. Data Intell.*, vol. 3, no. 3, pp. 350–365, Aug. 2022.

[9] M. R. S. Marques, T. Bianco, M. Roodnejad, T. Baduel, and C. Berrou, "Machine learning for explaining and ranking the most influential matters of law," in *Proc. 17th Int. Conf. Artif. Intell. Law*, Jun. 2019, pp. 239–243.

[10] M. Duque Lizarralde and H. A. Contreras, "The real role of AI in patent law debates," *Int. J. Law Inf. Technol.*, vol. 30, no. 1, pp. 23–46, Apr. 2022.

[11] B. Ait Benali, S. Mihi, A. Ait Mlouk, I. El Bazi, and N. Laachfoubi, "Arabic named entity recognition in social media based on BiLSTM-CRF using an attention mechanism," *J. Intell. Fuzzy Syst.*, vol. 42, no. 6, pp. 5427–5436, Apr. 2022.

[12] C. S. Ali, L. Hatab, and S. Abdennadher, "Enhancing deep learning with embedded features for Arabic named entity recognition," in *Proc. 13th Lang. Resour. Eval. Conf.*, 2022, pp. 4904–4912.

[13] N. Patil, A. Patil, and B. V. Pawar, "Named entity recognition using conditional random fields," *Proc. Comput. Sci.*, vol. 167, pp. 1181–1188, Jan. 2020.

[14] M. Ali, G. Tan, and A. Hussain, "Bidirectional recurrent neural network approach for Arabic named entity recognition," *Future Internet*, vol. 10, no. 12, p. 123, Dec. 2018.

[15] O. Kuru, O. A. Can, and D. Yuret, "CharNER: Character-level named entity recognition," in *Proc. COLING 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 911–921.

[16] X.-A. Wang and S. Wicker, "An artificial neural net Viterbi decoder," *IEEE Trans. Commun.*, vol. 44, no. 2, pp. 165–171, Feb. 1996.

[17] M. Jarrar, M. Khalilia, and S. Ghanem, "Wojood: Nested Arabic named entity corpus and recognition using BERT," in *Proc. 13th Lang. Resour. Eval. Conf.*, 2022, pp. 3626–3636.

[18] A. Mousa, I. Shahin, A. B. Nassif, and A. Elnagar, "Cascaded RBF-CBiLSTM for Arabic named entity recognition," in *Proc. Int. Conf. Commun., Comput., Cybersecurity, Informat. (CCCI)*, Nov. 2020, pp. 1–5.

[19] M. Affi and C. Latiri, "Arabic named entity recognition using variant deep neural network architectures and combinatorial feature embedding based on CNN, LSTM and BERT," in *Proc. 36th Pacific Asia Conf. Lang., Inf. Comput.*, 2022, pp. 302–312.

[20] X. Qu, Y. Gu, Q. Xia, Z. Li, Z. Wang, and B. Huai, "A survey on Arabic named entity recognition: Past, recent advances, and future trends," 2023, *arXiv:2302.03512*.

[21] R. Grishman and B. Sundheim, "Message understanding conference-6: A brief history," in *Proc. 16th Conf. Comput. Linguistics*, 1996, pp. 1–6.

[22] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.zip: Compressing text classification models," in *Proc. Int. Conf. Learn. Represent.*, 2017. [Online]. Available: https://arxiv.org/abs/1612.03651

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171–4186.

[24] N. Alsaaran and M. Alrabiah, "Arabic named entity recognition: A BERT-BGRU approach," *Comput., Mater. Continua*, vol. 68, no. 1, pp. 471–485, 2021.

[25] M. N. A. Ali, G. Tan, and A. Hussain, "Boosting Arabic named-entity recognition with multi-attention layer," *IEEE Access*, vol. 7, pp. 46575–46582, 2019.

[26] I. E. Bazi and N. Laachfoubi, "Arabic named entity recognition using deep learning approach," *Int. J. Electr. Comput. Eng.*, vol. 9, pp. 2025–2032, Jun. 2019.

[27] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-based model for Arabic language understanding," in *Proc. 4th Workshop Open-Source Arabic Corpora Process. Tools, With Shared Task Offensive Lang. Detection*, 2020, p. 9—15.

[28] N. F. Mohammed and N. Omar, "Arabic named entity recognition using artificial neural network," *J. Comput. Sci.*, vol. 8, no. 8, pp. 1285–1293, Aug. 2012.

[29] M. Gridac and H. Haddad, "Arabic named entity recognition: A bidirectional GRU-CRF approach," in *Proc. Int. Conf. Comput. Linguistics Intell. Text Process.* Cham, Switzerland: Springer, 2018, p. 264—275.

[30] D. Awad, C. Sabty, M. Elmahdy, and S. Abdennadher, "Arabic name entity recognition using deep learning," in *Proc. 6th Int. Conf. Stat. Lang. Speech Process.*, 2018, pp. 105–116.

[31] M. N. A. Ali and G. Tan, "Bidirectional encoder–decoder model for Arabic named entity recognition," *Arabian J. Sci. Eng.*, vol. 44, no. 11, pp. 9693–9701, Nov. 2019.

TAOUFIQ EL MOUSSAOUI received the master's degree from the Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco, in 2017, where he is currently pursuing the Ph.D. degree in artificial intelligence. His research interests include text mining, pattern recognition, information extraction, machine learning, and artificial intelligence.

LOQMAN CHAKIR is currently a Professor of computer science with the Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco. His research interests include artificial intelligence, text mining, pattern recognition, and optimization.

JAOUAD BOUMHIDI (Member, IEEE) is currently a Professor of computer science with the Faculty of Sciences Dhar El Mahraz, Sidi Mohamed Ben Abdellah University, Fez, Morocco. His research interests include artificial intelligence, multi-agent systems, and pattern recognition.