Classification of Moroccan Legal and Legislative Texts Using Machine Learning Models

Amina BOUHOUCHE[1], Mustapha ESGHIR[2], Mohammed ERRACHID[3]
Faculty of Sciences, Mohammed V University in Rabat, LabMIA-SI Rabat, Morocco[1,2]
Regional Center for Education and Formation Professions (CRMEF), Rabat, Morocco[3]

***Abstract*—Artificial intelligence tools have revolutionized many fields, bringing significant progress in automating tasks and solving complex problems. In this article, we focus on the legal domain, where the data to be processed are specific and in large quantities. Our study consists in carrying out an automatic classification of Moroccan legal and legislative texts in Arabic. In addition, we will conduct a series of experiments to evaluate the impact of stemming, class imbalance and the impact of data quantity on the performance of the models used. Given the specificity of the Arabic language, we used Natural Language Processing (NLP) tools adapted to this language. For classification, we worked with the following models: Support Vector Machine (SVM), Random Forests (RF), K Nearest Neighbors (KNN) and Naive Bayes (NB). The results obtained are very impressive, and the comparison of model outputs enriches the debate on specificities of each model.**

***Keywords*—Classification Arabic text; natural language processing; legal data; machine learning**

I. Introduction

Approaching textual data in the Arabic legal field presents dual challenges due to linguistic and legal complexities. The Arabic language is an extremely rich Semitic language both linguistically and culturally. It is the official language in over 20 countries and serves as one of the six official languages of the United Nations (UN). Textual data in the Arabic language presents a challenge due to the language's specificity and complexities. These complexities are evident in the distinction between Literary Arabic comprising Classical Arabic and Modern Standard Arabic (MSA), and Dialectal Arabic, which is highly diverse due to regional and national variations. Furthermore, the linguistic richness of the Arabic language imparts complexity at the syntactic, morphological, semantic, and orthographic levels [1]. Regarding the legal domain, which is a highly interesting field and happens to be the focus of our study, it has been approached using machine and deep learning techniques in diverse languages to explore prediction of judicial decisions [2], synthesis [3], and translation of legal texts [4]. However, not as much research has been done on the subject of Arabic. NLP tools, machine learning, and deep learning models are applied to legal texts to automate certain tedious tasks and enable professionals to focus on high-value-added tasks. In this paper, we have focused on applying these techniques to classify legal texts related to Moroccan regulations and legislation. Several models have been trained, including SVM, RF, KNN and NB classifiers. The aim of our study is to work with a wide range of data and to evaluate the behavior of the models in relation to different situations. We have chosen Moroccan legal and legislative texts covering different fields. These texts include the Civil Status Law, the Organic Finance Law, the Judicial Organization Law, the Traffic Code, the Family Code, the Criminal Procedure Law and the Commercial Code. Each text deals with distinct subjects and uses terms specific to those subjects. To our knowledge, this corpus has not been approached before.

The approach adopted in this work concerns the evaluation of the impact of stemming and data quantities on model performance. According to our information, previous work has not dealt with these aspects when approaching legal data in Arabic. By comparing the performance of models trained on stemmed and non-stemmed data, we can evaluate the behavior of each model in relation to the application or non-application of the stemming approach. In this study, we also analyzed the impact of class imbalance on model performance. Thus, we first evaluated the results obtained when training the models on a dataset containing minority classes, then we trained the models on a dataset where the classes are more balanced. Furthermore, we examined the effect of varying data quantity on the performance of these classifiers. By systematically varying the size of our dataset, we were able to observe how each model responded to different volumes of training data. This analysis provides valuable information on the scalability and robustness of each classifier.

To carry out this study, we have structured our article as follows: The characteristics and specifications of the Arabic language are discussed in Section II. Section III is devoted to the state of related research, while the Section IV outlines the methodology followed and details the experiments conducted in this study. The results obtained are presented and discussed in the Section V. In the conclusion we summarized the results obtained and proposed new perspectives for future work to tackle new legal tasks and enrich the current state of the art.

II. Arabic Language: Characteristics and Specificities

The Arabic language incorporates several particularities that differentiate it from Western languages. In addition to linguistic complexities, Arabic is distinguished by its writing direction, from right to left, and the use of diacritical marks that determine the phonological meaning of a word [5]. The language also includes additional characters, such as "إ، آ، أ، ء، ئ، ؤ، ى" and "ة", whose basic form are the Arabic letters "ا" and "ه" respectively. Furthermore, the syntax of the Arabic language differs from that of Western languages, especially concerning word order. For example, in the sentence "أصدرت المحكمة الحكم" where the direct translation is "issued

the court the judgment", the order of the subject and the verb is reversed compared to their order in the English language, where it is "the court issued the judgment". A deep understanding of grammatical and syntactical structures is essential for an enhanced mastery of the language and to ensure a smooth translation between multiple languages.

Morphologically, a trilateral root has the ability to generate multiple derivatives by adding prefixes, suffixes, or modifying vowels. For instance, the trilateral root "حكم" meaning "to judge" can give rise to various words related to the field of judgment and law. These include "حُكْم" (hukm) meaning "judgment", "حكمة" (hik-ma) meaning "wisdom" and "محكمة" (mah-ka-ma) meaning "court". Derivatives also provide the opportunity to express various nuances, such as gender, with "حاكم" (ha-kim) meaning male judge and "حاكمة" (ha-ki-ma) meaning female judge, as well as the dual form with "حاكمان" (ha-ki-ma-an) or plurals "حكام" (hu-kka-m), and so forth.

Regarding the semantic aspect, there are words that are written in the same way but differ in terms of meaning and pronunciation. Understanding the correct sense and determining the correct pronunciation are made possible by the context of use. The term "شهادة" for example, can have various meanings such as "testimony" or "certificate," and only the context of the sentence or surrounding discourse allows for determining the specific meaning.

The spelling of Arabic letters is also very specific. The shape of the letters changes according to their position in the word. The letter "ع" for example, has four forms: one form at the beginning of the word as in "عقد", a middle form as in the word "تعلم", and two forms at the end of the word as in the words "استماع" and "بيع."

III. Related Works

The use of machine learning tools has gained momentum in the approach to judicial data. Aleteras et al [2] conducted the first study to combine machine learning and NLP tools to predict judicial decisions regarding the violation of human rights convention articles. This study used the Support Vector Machine (SVM) classifier and served as a reference for several subsequent works, particularly those conducted by Sulea et al. [6] and Liu and Chen [7]. In 2017, Katz et al. [8] presented a study in a generalized, consistent, and out-of-sample framework using the Random Forest classifier to predict US Supreme Court behavior. The promising outcomes in predicting judicial case have piqued the interest of researchers worldwide. For instance, Walt et al. [9] applied the Naive Bayes classifier to German jurisdiction, particularly German tax law cases. Likewise, Virtucio et al. [10] utilized Aletras's method to predict decisions of the Philippine Supreme Court. On the other hand, deep learning models were initially experimented by the Chinese for predicting legal judgments as a multitask problem. Luo et al.'s study [11] focused on simultaneously modeling relevant law extraction and charge prediction. Zhong et al. [12] opted for topological learning for predicting three dependent subtasks: applicable law articles, charges, and penalties. On the other hand, Long et al. [13] developed a model based on legal reading comprehension to express complex semantic interactions between factual descriptions, pleadings, and law articles. Ye et al.'s [14] work revolves around a Seq2Seq model to combine charge prediction and court opinion generation. Meanwhile, Hu et al [15] designed a model focused on predicting infrequent charges.

The deep learning models have attracted the attention of other researchers who have worked on datasets from different courts and jurisdictions. Kowsrihawat et al.[16] utilized the BI-GRU model with an attention mechanism to analyze a corpus of criminal cases from the Thai Supreme Court and predict the guilt or innocence of an accused. Chalkidis et al. [17] also worked on data from the European Court of Human Rights (ECHR) for prediction purposes. They tested several models, including SVM BOW, BIGRU-Attention, Hierarchical Attention Network, Label-Wise Attention Network, and HIERARCHICAL-BERT. Their analysis showed that deep learning models perform better. For their part, Malik et al. [18] designed a prediction system for Indian Supreme Court cases, with the capability of explaining the obtained predictions. For prediction, the authors explored classical, sequential, and hierarchical models as well as transformer-based models.

The application of machine learning methods to judicial data in various languages has led to great interest in the exploration of Arabic court data. Study [19] focused on predicting Arabic judgments from the Errachidia court in Morocco, specifically targeting accident cases. Deployed machine learning models included Linear Regression, Random Forest, and Decision Trees, achieving encouraging results with 91% accuracy for Random Forest. Shamma et al. [20] addressed information extraction from Arabic legal documents, applying Arabic-specific natural language processing techniques and machine learning models like SVM, Decision Trees, and KNN. In study [21], researchers examined the prediction of verdicts, legal articles, and probabilities of judgment, focusing on child custody and marriage annulment cases in the Arabic language. They experimented with TF-IDF and word2vec representation techniques and used various machine and deep learning models including SVM, Logistic Regression (LR), LSTM, and BiLSTM. Article [22] introduces TaSbeeb, an innovative decision support tool for the Saudi judicial system. Designed for Arabic language, it retrieves judicial reasoning and performs multiclass classification of legal cases. The study proposes deep models Att-GRU, BiLSTM, and BiGRU with stacking approach, both homogeneous and heterogeneous, along with an Arabic judicial language model called Jud_RoBERTa.

In general, the research works that have approached the legal field in Arabic remain limited. Table I below presents some of these works. Most of them focus on judgments or judicial decisions in narrow fields. In addition, this type of data can sometimes lack precision, which can have an impact on the final decision given by the models. Another limitation encountered by the authors lies in the quantities of data available, whereas the models require adequate and representative quantities of data for better training. In our study we worked with legal and legislative data covering seven different domains. In addition, we focused on the impact of data on model performance by changing the amounts of data used and evaluating the impact of class imbalance and

stemming on model performance.

TABLE I. STUDIES ON THE APPLICATION OF ARTIFICIAL INTELLIGENCE APPROACHES TO ARABIC LEGAL DATA

Articles	Experiment	Data
[19]	Predicting the outcome of accident cases	Accident cases issued by the Errachidia court in Morocco
[20]	Information extraction from Arabic legal documents	Appeal cases for leased premises lawsuits
[21]	Predicting judgments and legal articles or proof as well as probabilities of expected results	Marriage annulment and child custody cases
[22]	Development of a decision support system for the Saudi judicial court	Family and personal status cases
[25]	Classification of Arabic legal documents	Judgments relating to real estate and road traffic issued by the Moroccan Supreme Court

IV. METHODOLOGY

The categorization of textual data is one of the fundamental pillars of much research that has explored the contributions of artificial intelligence tools to the legal field. This importance stems from the need to optimize time and resources, particularly in view of the large quantities of data to be processed, including judgments, legal texts and investigation reports. The classification of this type of data provides invaluable assistance to legal professionals, facilitating access to relevant information, comparative analysis between different laws or regulations, and research into previous similar cases.

In this study, we focused on Arabic-language legal data relating to Moroccan regulations. To classify this text data, it is important to follow a clearly defined process, including data collection and pre-processing, vector representation of the texts, model application and performance evaluation (Fig. 1).

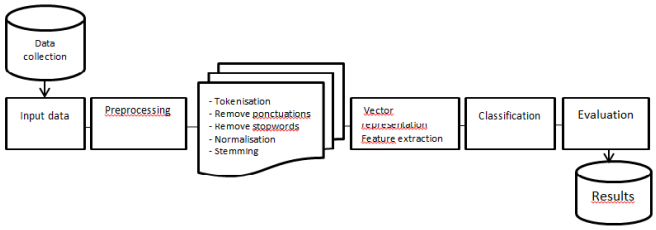

Fig. 1. Classification process for legal and legislative texts.

A. Data Collection

The dataset represents the input for algorithms and the foundation of learning. The models' performance depends entirely on the quality and quantity of the chosen corpus.

In this study, we have gathered seven Moroccan legislative and legal texts, detailed in (Table II). These texts include articles related to the Civil Status Law, the Organic Finance Law, the Judicial Organization Law, the Traffic Code, the Family Code, the Criminal Procedure Law and the Commercial Code. The number of articles varies from one legal text to another, and similarly, the number of words in these articles varies from one article to another. The total number of articles is 2510. The Commercial Code and the Criminal Procedure Law are the most extensive, with 798 and 757 articles, respectively. Since the assignment of articles to a class is determined by the designation of the appropriate legal text, the distribution of classes in our dataset is not balanced.

TABLE II. OVERVIEW OF STUDIED DATA

Legal text	Number of articles
"القانون رقم " 36.21 "المتعلق بالحالة المدنية"	59
"القانون التنظيمي رقم " 130-13 "قانون المالية"	70
" القانون رقم " 38.15 "المتعلق بالتنظيم القضائي"	111
"مدونة السير على الطرق"	318
"مدونة الأسرة"	400
"القانون رقم " 22.01 "المتعلق بالمسطرة الجنائية"	757
"مدونة التجارة"	798

B. Preprocessing

Given our interest in the Arabic legal domain, we have examined NLP tools applied to Arabic-language data. Below, we present the main pre-processing steps required to prepare textual data and make it easy and suitable to handle.

1) Tokenization: Segmentation of text into meaningful linguistic entities.

2) Removing stop words: Elimination of words that don't provide significant information, using the Arabic stop words list.

3) Normalization: Normalization involves writing words in a specific form, bringing letters back to their base form. For example, converting "hamza" in all its forms "إ، آ، أ، ء، ئ، ؤ، ى" to "alef" "ا".

4) Stemming: The process of reducing words to their base form. Among the most frequently employed stemmers for the Arabic language, there are Khoja stemmer and ISRI Stemmer [23]. In this study, we used ISRI Stemmer.

	Sentence1	Sentence 2
Input text	يأمر قاضي التحقيق بتبليغ الشكاية إلى وكيل الملك أو الوكيل العام للملك لتقديم ملتمساته.	تشتمل النفقات المتعلقة بالدين العمومي على النفقات من فوائد وعمولات والنفقات المتعلقة باستهلاكات الدين المتوسط والطويل الأجل
Pre-processed text without stemming step	يامر قاضي التحقيق بتبليغ الشكايه وكيل الملك الوكيل العام للملك لتقديم ملتمساته	تشتمل النفقات المتعلقه بالدين العمومي النفقات فوائد وعمولات والنفقات المتعلقه باستهلاكات الدين المتوسط والطويل الاجل
Pre-processed text with stemming step	امر قضي حقق بلغ شكي وكل ملك وكل عام ملك قدم ملتمساته	شمل نفق تعلق دين عمم نفق فئد عمل نفق تعلق باستهلاك دين توسط طول اجل

Fig. 2. Examples of sentences illustrating the pre-processing of legal and legislative texts.

Fig. 2 illustrates the pre-processing of two sentences, the first extracted from the Criminal Procedure Law and the second from the Organic Finance Law. The results of the pre-processing show that the tools applied have succeeded to a

certain extent in approximating the Arabic text. However, the difficulties encountered include the case of attached words, where it is difficult to identify stop words. We take as an example the word "باستهلاك" composed of two elements "ب" which is a preposition and the word "استهلاك" meaning "consumption". In principle the stop word should be removed. And the term "استهلاك" should undergo stemming. By analyzing the output of the preprocessing with stemming and without stemming step, we noticed that the tool used has succeeded in reducing words to their roots, such as transforming "التحقيق" into "حقق" and "الوكيل" into "وكل" . However the stemming of some words failed. For example the word "فوائد" was rendered to a non-existent word in the Arabic language "فئد" and the words "ملتمساته" and "استهلاك" were not stemmed.

C. Vector Representation Methods

To ensure that textual data is readable and interpretable by algorithms, it must be converted into vector representations. The methods commonly used for this conversion are as follows:

1) BOW: Vector representation of the document that associates each word with its frequency of occurrence.

2) TFIDF: A representation that emphasizes the importance of a word in a document relative to its occurrence in the entire corpus.

$$TF - IDF(term) = TF(term) \times log(\frac{N}{DF(term)}) \quad (1)$$

TF: the number of times the term occurs in a document

DF: the number of documents in which the term is present

N : total number of documents

3) Word embedding: Representations that capture semantic relationships between words. These representations are generated by widely used models such as Word2Vec, GloVe and FastText [24], initially designed for Western languages and applied to the Arabic language.

D. Model Application

Machine learning models are selected according to the nature of the data and the specific objectives of the task. In this study, we chose to use four of the best-known classical classification models: SVM, KNN, Random Forest and Naive Bayes, based on the scikit-learn implementation.

1) Support Vector Machine (SVM): A supervised learning method employed for classification tasks. SVM focuses on choosing the optimal hyperplane to maximize margin separation and incorporates the Kernel function. This function facilitates the projection of data into a higher-dimensional space to handle non-linear classification challenges. The SVM classifier has proven to be very interesting for text classification [25]. Its application to Arabic legal data has been conducted for several tasks, including classification [21] and information retrieval [20].

2) K Nearest Neighbors (KNN): A straightforward and non-parametric classification model based on neighborhood principle. It depends on selecting a distance metric to calculate the distance between instances. The classification of a new instance is decided by the majority class of the k nearest data points in the training set. Although this technique is widely used in text classification, its application to Arabic legal data remains limited, among the few works that have used it, we can cite the study by Ikram and Chakir [25] to classify Moroccan court rulings.

3) Naive bayes: A probabilistic model grounded in Bayes' theorem with conditional independence assumption. This means that the values of predictors of a given class are conditionally independent of each other.

$$P(c_k \mid x_i) = \frac{P(c_k) \times \prod_{j=1}^{n} P(x_{ij} \mid c_k)}{\sum_{s=1}^{K} P(c_s) \prod_{j=1}^{n} P(x_{ij} \mid c_s)} \quad (2)$$

K: number of class modalities
$C = (c_1, c_2, ...c_k, ..., c_K)$: the class variable
n: the number of features or explanatory variables

$x_i = (x_{i1}, ...x_{ik}...x_{in})$: vector of explanatory variables

$P(c_k \mid x_i)$: the probability that the document represented by x_i, belongs to the class c_k.

$P(x_{ij} \mid c_k)$: The likelihood of the feature x_{ij} conditioned on the class c_k

P(c_k): represents the prior probability associated with the class c_k

The target class is the class for which the probability P($c_k \mid x_i$) is maximum:

$$c_k = ArgMax(P(c_k) \prod_{j=1}^{n} P(x_{ij} \mid c_k) \quad (3)$$

The Naive Bayes classifier has been widely used in various research exploring the legal domain across multiple languages and it is also employed in the classification of legal texts in Arabic [25][26].

4) Random forest: An ensemble machine learning model that combines decision trees of CART-type through aggregation. Each tree is built from a bootstrap sample of the training set, using a random subset of features. The model's prediction is established through a majority vote from all the trees. Random Forest is a widely used model for text data analysis. It has been employed by researchers such as Katz [8], Liu and Chen [7] and others to approach judicial data. Concerning the Arabic language, this model has been applied to tasks such as isolated Arabic character recognition [27] and legal document classification [25].

E. Evaluation

The performance of classification and prediction models is evaluated through various metrics. This approach aims to analyze the quality of results, the overall efficiency of models, and to understand their behavior in a variety of situations. The most commonly used measurements in this context include:

Accuracy: Proportion of correctly classified cases to the total number of cases.

$$Accuracy = \frac{TP + TN}{Total} \tag{4}$$

Recall: Proportion of cases correctly assigned to a class i to all cases in that class.

$$Rappel = \frac{TP}{TP + FN} \tag{5}$$

Precision: Proportion of cases correctly assigned to class i compared to all cases predicted to be assigned to class i.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

F1 score: Harmonic mean of precision and recall.

TP: True positive

TN: True negative

FP: False positive

FN: False negative

These performance measures are typically used for binary classification, however, there are methods to generalize them to include multiple classification:

1) Micro approach: The global metrics are calculated from the set contributions of all classes. The TP, TN, FP and FN of all classes are combined to calculate a single accuracy, precision, recall and F1-score.

2) Macro approach: consists of calculating the metrics for each class individually and then calculating an average of all the classes for each measure. These averages can be arithmetic when the classes are balanced and weighted averages when the classes are unbalanced.

F. Experimentation

Due to our interest in applying machine learning tools in the legal field, we conducted a series of experiments to evaluate the impact of stemming, class imbalance and data quantity on model performance.

Our data corpus consists of articles from seven legal and legislative texts (Table II). The designation of the appropriate text constitutes the class assignment for the articles, resulting in a total of seven classes. To ensure the quality of our analysis, we first prepared our data using preprocessing techniques mentioned above. We then applied our models and evaluate the performance obtained. We conducted five distinct experiments to assess the performance of our models in different scenarios. First, we trained classification models on the entire corpus and evaluated performance (Table III). This first experiment served as a basis of comparison for some of the subsequent experiments. In the second experiment, we drew inspiration from the study by Taghva et al. [23]to evaluate the effectiveness of stemming on the performance of our models. In this experiment, we eliminated the stemming step and compared the results (Table IV) with those of the first experiment. Inspired by Sulea et al's study [6] of French-language court decisions, we came up with the idea of assessing the impact of class imbalance and the presence of minority classes when exploring Arabic-language legal documents. Thus, in a third experiment, we assessed the impact of minority classes on model performance by eliminating classes with fewer than 200 articles and retraining our models (Table V). This reduced the corpus to four classes, and the results were compared with those from the first experiment. In the fourth experiment, we evaluated the impact of class imbalance on model performance (Table VI). To do this, we worked with the four classes from the previous experiment and retained 300 articles per class. Finally, to assess the impact of data quantity, we conducted a fifth experiment where we reduced the number of articles to 50 per class(Table VII).

V. Discussion

Our study focuses on the analysis of Moroccan legislative and legal texts using natural language processing tools specifically designed for the Arabic language. We conducted classification experiments on this corpus of data using four classifiers: SVM, KNN, Random Forest and Naive Bayes.

TABLE III. Model Performance with Stemming Step

Classifier	Accuracy	Precision	Recall	F1 score
SVM	94.58	94.9	94.58	94.45
KNN	92.17	92.56	92.17	92.26
Random Forest	91.16	91.91	91.16	90.75
Naive Bayes	92.17	92.72	92.17	91.82

TABLE IV. Model Performance without Stemming Step

Classifier	Accuracy	Precision	Recall	F1 score
SVM	92.57	93.16	92.57	91.86
KNN	92.57	93.16	92.57	92.77
Random Forest	91.97	92.4	91.97	91.75
Naive Bayes	95.38	95.6	95.38	95.19

TABLE V. Model Performance Without Minority Classes

Classifier	Accuracy	Precision	Recall	F1 score
SVM	95.82	95.89	95.82	95.83
KNN	96.26	96.28	96.26	96.26
Random Forest	94.95	94.98	94.95	94.95
Naive Bayes	93.41	93.48	93.41	93.41

TABLE VI. Model Performance on Balanced Classes (300 Articles Per Class)

Classifier	Accuracy	Precision	Recall	F1 score
SVM	96.25	96.38	96.25	96.25
KNN	95.42	95.51	95.42	95.42
Random Forest	93.33	93.54	93.33	93.34
Naive Bayes	95.83	96.04	95.83	95.88

TABLE VII. MODEL PERFORMANCE ON SMALL BALANCED DATA (50 ARTICLES PER CLASS)

Classifier	Accuracy	Precision	Recall	F1 score
SVM	92.5	95.00	92.5	93.01
KNN	95	95.23	95	95.02
Random Forest	92.5	93.75	92.5	92.72
Naive Bayes	97.50	97.86	97.50	97.55

The graph in Fig. 3 summarizes the accuracy of all the models across the five experiments detailed above.

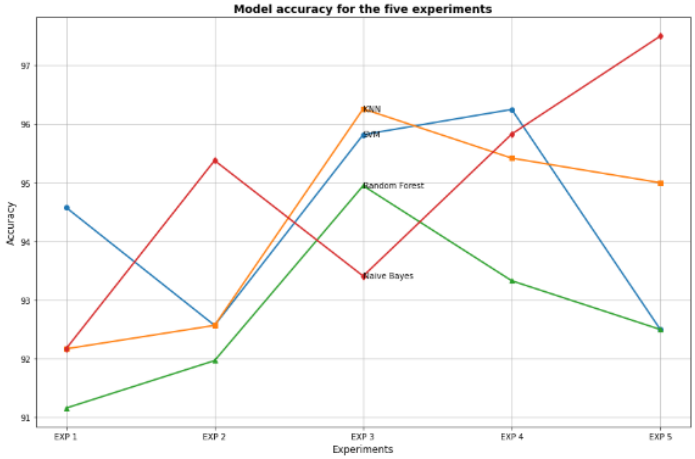

Fig. 3. Model accuracy for the five experiments.

The results of the first experiment (Table III), carried out on the whole corpus, are very promising, exceeding 90% for all models. The SVM classifier achieved the best performance across all metrics, with an accuracy of 94.58% and an F1 score of 94.45%. This finding is consistent with that of Liu and Chen [7], who worked on English-language data from the European Court of Human Rights and confirmed that SVM was the most efficient classifier.

In the second experiment, where we omitted the stemming step (Table IV), we observed that the classifiers behaved differently. The performance of the SVM model decreased, achieving 92.57% accuracy, while the performance of KNN, Random Forest and Naive Bayes has improved, achieving 92.57%, 91.97% and 95.38% accuracy respectively. This difference in behavior is explained by the fact that each classification algorithm has its own characteristics and assumptions. This experiment showed that the KNN, Random Forest and Naive Bayes algorithms are sensitive to the changes introduced by stemming, and are less tolerant of the loss of information it generates. These models are better suited to exploiting the lexical diversity of non-stemmed data, giving them a better understanding of nuances and relationships between words. On the other hand, the SVM classifier, whose performance decreased when trained on non-stemmed words, has been shown to be sensitive to the noise introduced by morphological variations. In addition, the use of stemmed words, allowed the SVM to benefit from the dimensionality reduction introduced by stemming.

The third experiment, in which the minority classes were removed (Table V), revealed an improvement in the performance of all models, with an accuracy of 95,82% for SVM, 96,26% for KNN, 94,95% for Random Forest and 93,41% for Naive Bayes. This suggests that the presence of minority classes was limiting the models' ability to train effectively and identify underlying patterns in the data. By eliminating these minority classes, the models were able to focus on more representative classes, reducing noise and imbalances in the training data. This simplification of the dataset enabled the algorithms to generalize better and produce more accurate and consistent results.

The fourth experiment assessed the impact of class balance, retaining 300 articles per class for the four classes selected in the previous experiment. This approach ensured a fair distribution of data between classes, reducing the bias introduced by class imbalances. By balancing the number of articles per class, the models were able to train on representative datasets, achieving interesting levels of accuracy: 96.25% for SVM, 95.42% for KNN, 93.33% for Random Forest and 95.83% for Naive Bayes (Table VI).
These last two experiments demonstrate the importance of class balance in the machine learning model training process, contributing to more robust performance.

By reducing the amount of data in the fifth experiment (Table VII), we observed a decrease in the performance of the SVM, KNN and Random Forest classifiers compared with the fourth experiment. The accuracy of these classifiers is reduced to 92.5%, 95% and 92.5% respectively. This decrease can be attributed to various factors inherent to these algorithms, mainly the careful tuning of hyperparameters. This tuning process is often made more difficult when the available data is limited. In addition, the Random Forest model, for example, requires a sufficient amount of data for each tree to learn meaningful and diverse patterns. By contrast, the Naive Bayes classifier performed better with small amounts of data. This performance is attributed to the simplicity of the model and its conditional independence assumption, which assumes that each feature contributes independently to the probability of each class.

VI. CONCLUSION

In this study, we were interested in the application of natural language processing and machine learning tools to classify Moroccan legal and legislative data. Given that our datasets are in Arabic, this adds a second level of complexity to our study. The texts under study include the Law on Civil Status, the Organic Law on Finance, the Law on Judicial Organization, the Traffic Code, the Family Code, the Law on Criminal Procedure and the Commercial Code. To the best of our knowledge, this data has not been covered before. Our task was to train the SVM, KNN, Random Forest and Naive Bayes models to assign each article to the appropriate class. In addition to this classification task, we evaluated the impact of stemming, class imbalance and data quantity on the behavior of the models used. In general, our experiments have shown that models perform better with balanced classes, but react differently to stemming and data quantity. Although these classical models generally perform well, it's important to note that they remain sensitive to word variations and the size of the training datasets. In future work, we will explore new perspectives by investigating other models, including deep

learning and language models, in order to tackle new legal tasks.

ACKNOWLEDGMENT

This work was supported by the Ministry of Higher Education, Scientific Research and Innovation, the Digital Development Agency (DDA) and the CNRST of Morocco (Alkhawarizmi/2020/25).

REFERENCES

[1] Shaalan, K., Siddiqui, S., Alkhatib, M., & Abdel Monem, A. (2019). Challenges in Arabic natural language processing. In Computational linguistics, speech and image processing for arabic language (pp. 59-83).

[2] Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., & Lampos, V. (2016). Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. PeerJ computer science, 2, e93.

[3] Anand, D., & Wagh, R. (2022). Effective deep learning approaches for summarization of legal texts. Journal of King Saud University-Computer and Information Sciences, 34(5), 2141-2150.

[4] Wiesmann, E. (2019). Machine translation in the field of law: A study of the translation of Italian legal texts into German. Comparative Legilinguistics, 37(1), 117-153.

[5] Darwish, K., Habash, N., Abbas, M., Al-Khalifa, H., Al-Natsheh, H. T., Bouamor, H., ... & Mubarak, H. (2021). A panoramic survey of natural language processing in the Arab world. Communications of the ACM, 64(4), 72-81.

[6] Şulea, O. M., Zampieri, M., Vela, M., & van Genabith, J. (2017, September). Predicting the Law Area and Decisions of French Supreme Court Cases. In Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017 (pp. 716-722).

[7] Liu, Z., & Chen, H. (2017, November). A predictive performance comparison of machine learning models for judicial cases. In 2017 IEEE Symposium series on computational intelligence (SSCI) (pp. 1-6). IEEE.

[8] Katz, D. M., Bommarito, M. J., & Blackman, J. (2017). A general approach for predicting the behavior of the Supreme Court of the United States. PloS one, 12(4), e0174698.

[9] Waltl, B., Bonczek, G., Scepankova, E., Landthaler, J., & Matthes, F. (2017). Predicting the outcome of appeal decisions in Germany's tax law. In Electronic Participation: 9th IFIP WG 8.5 International Conference, ePart 2017, St. Petersburg, Russia, September 4-7, 2017, Proceedings 9 (pp. 89-99). Springer International Publishing.

[10] Virtucio, M. B. L., Aborot, J. A., Abonita, J. K. C., Avinante, R. S., Copino, R. J. B., Neverida, M. P., ... & Tan, G. B. A. (2018, July). Predicting decisions of the philippine supreme court using natural language processing and machine learning. In 2018 IEEE 42nd annual computer software and applications conference (COMPSAC) (Vol. 2, pp. 130-135). IEEE.

[11] Luo, B., Feng, Y., Xu, J., Zhang, X., & Zhao, D. (2017, September). Learning to Predict Charges for Criminal Cases with Legal Basis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (pp. 2727-2736).

[12] Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., & Sun, M. (2018). Legal judgment prediction via topological learning. In Proceedings of the 2018conference on empirical methods in natural language processing (pp. 3540-3549).

[13] Long, S., Tu, C., Liu, Z., & Sun, M. (2019). Automatic judgment prediction via legal reading comprehension. In Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18 (pp. 558-572). Springer International Publishing.

[14] Ye, H., Jiang, X., Luo, Z., & Chao, W. (2018, June). Interpretable Charge Predictions for Criminal Cases: Learning to Generate Court Views from Fact Descriptions. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (pp. 1854-1864).

[15] Hu, Z., Li, X., Tu, C., Liu, Z., & Sun, M. (2018, August). Few-shot charge prediction with discriminative legal attributes. In Proceedings of the 27th international conference on computational linguistics (pp. 487-498).

[16] Kowsrihawat, K., Vateekul, P., & Boonkwan, P. (2018, October). Predicting judicial decisions of criminal cases from Thai Supreme Court using bi-directional GRU with attention mechanism. In 2018 5th Asian Conference on Defense Technology (ACDT) (pp. 50-55). IEEE.

[17] Chalkidis, I., Androutsopoulos, I., & Aletras, N. (2019, July). Neural Legal Judgment Prediction in English. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 4317-4323).

[18] Malik, V., Sanjay, R., Nigam, S. K., Ghosh, K., Guha, S. K., Bhattacharya, A., & Modi, A. (2021, August). ILDC for CJPE: Indian Legal Documents Corpus for Court Judgment Prediction and Explanation. arXiv preprint arXiv:2105.13562

[19] Haidar, A., Ahajjam, T., Zeroual, I., & Farhaoui, Y. (2022). Application of machine learning algorithms for predicting outcomes of accident cases in Moroccan courts. Indonesian Journal of Electrical Engineering and Computer Science, 26(2), 1103-1108.

[20] Shamma, S. A., Ayasa, A., & Yahya, A. (2020, October). Information extraction from arabic law documents. In 2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT) (pp. 1-6). IEEE.

[21] Abbara, S., Hafez, M., Kazzaz, A., Alhothali, A., & Alsolami, A. (2023). ALJP: An Arabic Legal Judgment Prediction in Personal Status Cases Using Machine Learning Models. arXiv preprint arXiv:2309.00238.

[22] Almuzaini, H. A., & Azmi, A. M. (2023). TaSbeeb: A judicial decision support system based on deep learning framework. Journal of King Saud University-Computer and Information Sciences, 35(8), 101695.

[23] Taghva, K., Elkhoury, R., & Coombs, J. (2005, April). Arabic stemming without a root dictionary. In International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II (Vol. 1, pp. 152-157). IEEE.

[24] Dharma, E. M., Gaol, F. L., Warnars, H. L. H. S., & Soewito, B. E. N. F. A. N. O. (2022). The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (cnn) text classification. J Theor Appl Inf Technol, 100(2), 31.

[25] Ikram, A. Y., & Chakir, L. O. Q. M. A. N. (2019, October). Arabic text classification in the legal domain. In 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS) (pp. 1-6). IEEE.

[26] Jasim, K., Sadiq, A. T., & Abdullah, H. S. (2019, December). A framework for detection and identification the components of arguments in Arabic legal texts. In 2019 First International Conference of Computer and Applied Sciences (CAS) (pp. 67-72). IEEE.

[27] Rashad, M., & Semary, N. A. (2014). Isolated printed Arabic character recognition using KNN and random forest tree classifiers. In Advanced Machine Learning Technologies and Applications: Second International Conference, AMLTA 2014, Cairo, Egypt, November 28-30, 2014. Proceedings 2 (pp. 11-17). Springer International Publishing.