On Integrating and Classifying Legal Text Documents

Alexandre Quemy[1,2] [0000−0002−5865−6403]
Robert Wrembel[2] [0000−0001−6037−5718]

[1] IBM, Cracow Software Lab, Poland
[2] Poznan University of Technology, Poznań, Poland
[1] aquemy@pl.ibm.com
[2] robert.wrembel@cs.put.poznan.pl

Abstract. This paper presents an exhaustive and unified dataset based on the European Court of Human Rights judgments since its creation. The interest of such database is explained through the prism of the researcher, the data scientist, the citizen and the legal practitioner. Contrarily to many datasets, the creation process, from the collection of raw data to the feature transformation, is provided under the form of a collection of fully automated and open-source scripts. It ensures reproducibility and a high level of confidence in the processed data, which is some of the most important issues in data governance nowadays. A first experimental campaign is performed to study some predictability properties and to establish baseline results on popular machine learning algorithms. The results are consistently good across the binary datasets with an accuracy comprised between 75.86% and 98.32% for a micro-average accuracy of 96.44%.

Keywords: legal text document integration · text analytics · text document classification

1 Introduction

Machine learning (ML) algorithms are used in multiple domains (e.g., sales, healthcare, production), as they build prediction models of acceptable quality and yet explainable. However, the application of ML to the legal domain so far has received little attention from research communities [4, 17], but the need of ML solutions to support judicial decision is slowly becoming recognized (e.g., study programs combining artificial intelligence and law at Duke University (USA), Swansea University (UK), Maastricht Unviersity (NL) [1]).

Applying ML algorithms in the law domain is challenging. First, the legal domain is a messy concept [21] that intrinsically creates some of the most challenging problems for the ML research community including: gray areas of interpretation, many exceptions, non-stationarity, presence of deductive and inductive reasoning, non-classical logic, multiple and complex legal rules, as well as semantic complexity of legal acts. Second, there are few large open repositories of legal cases, with clean, adequately structured data. As a consequence, it is challenging to verify ML algorithms on legal data. From the set of ML algorithms [12], classification is a primary technique for building prediction models in the legal domain [15].

There exist few initiatives to provide open data repositories on judicial cases, including the recent one in Australia (AI for Law Enforcement and Community Safety that supports automated classification of online child exploitation material) and Singapore (Intelligent Case Retrieval System that enables retrieval of relevant precedent cases by means of artificial intelligence tools) [23]. From the available judicial repositories, the most known ones include: the *Supreme Court of the United States* and the *European Court of Human Rights*. Even though these corpora of legal cases are available, multiple information are missing and an access interface to these repositories is limited (cf., Section 2). Moreover, the content of these repositories has to be pre-processed before ML algorithms are run on them, as incomplete and inadequately prepared data for ML algorithms strongly impact a quality of built prediction models [8, 9, 18]. Recently, we analyzed and experimentally showed that the way data are pre-processed for classification algorithms impacts the quality of classificators [6, 7].

The aforementioned observations motivated us to build and make available an open, exhaustive, and unified data repository, called *ECHR-DB*, about legal cases from the European Court of Human Rights. The repository is accompanied by a comprehensive processing pipeline, neatly documented and supported by rich metadata, to provide reusability, repeatability of experiments, and manageability. In details, the paper **contributes** the following:

1. A **benchmark suit** for ML algorithms in the law domain, based on the European Court of Human Rights. The benchmark is composed of: (1) the *ECHR-DB* repository that stores almost all cases judged by the European Court of Human Rights since its creation, cleaned and transformed to ease the exploration by ML algorithms and (2) 13 standard ML algorithms that can be immediately run on *ECHR-DB*.
2. The whole **ETL and data transformation pipeline** used to generate the benchmark suit, available as an open-source project. As a consequence, the whole data ingestion, transformation, and cleaning processes can be repeated, revised, and extended.
3. A comparison of **13 standard machine learning algorithms** for classification with regards to several performance metrics. These results provide a baseline for future studies and provide some insights about the interest of some types of features to predict justice decisions.

The paper is organized as follows. Section 2 presents related work on analytics in the legal domain. Section 3 outlines the *ECHR-DB* repository. The process of creating the repository is discussed in Section 4. Section 5 reports the experiments on the repository. Section 6 summarizes and concludes the paper.

2 Related Work

Predicting the outcome of a justice case is challenging, even for the best legal experts. As shown in [22], 67.4% and 58% accuracy was achieved, respectively for the judges and the whole case decision, using cases from the Supreme Court of the United States. Using crowds, the *Fantasy Scotus*[1] project reached 85.20% and 84.85% correct predictions, respectively. In [15], the authors proposed to apply an SVM-based classificator and they were able to correctly predict about 75% of the cases.

A success of research in ML for the legal domain depends on the availability of large datasets of legal cases with judicial decisions. There are a few open data repositories of judicial cases available. The most known ones include: the *SCOTUS* repository[2] of the *Supreme Court of the United States* and the *HUDOC* database[3] of the *European Court of Human Rights*. *SCOTUS* is composed of structured data (in a tabular format) about every case since the creation of the court but it lacks textual information about decisions. *HUDOC* contains all legal cases with judgments. However, its interface has some flaws, e.g., it does not offer any API to allow to access several documents at once and case documents are not unified in the way that they could offer tabular and natural language data. In other words, despite its public availability, the data are hard to retrieve and to work with.

The prediction of the *Supreme Court of the United States* has been widely studied, notably through the *SCOTUS* repository [11, 14, 10]. To the best of our knowledge, the only predictive models that used the content of *HUDOC* were reported in [2, 15]. The data used in [2] are far from being exhaustive: only 3 articles considered (3, 6 and 8) with respectively 250, 80 and 254 cases per article. Using SVM with linear kernel, the authors achieved 79% accuracy to predict the decisions of the European Court of Human Rights. SVM is also used in [15] to reach an overall of 75% accuracy on judgment documents up to September 2017. In [17], the author outlined some practical problems in the field of legal analytics, notably the prediction and the justification problem.

New studies tend to suggest that there will always be a limit in reasoning systems to handle new cases presenting novel situations [5], which emphasize the interest for data-centric methods, hence the need for *large and adequate sets of legal data* (mainly cases and their justifications) available to researchers and practitioners. Such datasets should be equipped with: (1) a user-friendly interface to access and analyze the data and (2) rich metadata to offer means for browsing the content of a repository and to tune ML algorithms. Unfortunately, the aforementioned databases do not fully meet these requirements. This observation motivated us to start the project on building an open European Court of Human Rights repository (*ECHR-DB*).

[1] `https://fantasyscotus.lexpredict.com/`
[2] `http://scdb.wustl.edu/`
[3] `https://hudoc.echr.coe.int/eng`

3 ECHR-DB in Brief

The *ECHR-DB* repository aims at providing exhaustive and high-quality database for diverse problems, based on the European Court of Human Rights documents from *HUDOC*. The main objectives of this project are as follows: (1) to draw the attention of researchers on this domain that has important consequences on the society and (2) to provide a similar and more complete database for the European Union as it already exists in the United States, notably because the law systems are different in both sides of the Atlantic.

ECHR-DB is guided by three core values: **reusability**, **quality** and **availability**. To reach those objectives:

- each version of the datasets is carefully versioned and publicly available, including the intermediate files,
- the integrality of the process and files produced are careful documented,
- the scripts to retrieve the raw documents and to build the datasets from scratch are open-source and carefully versioned to maximize reproducibility and trust,
- no data is manipulated by hand at any stage of the creation process to make it fully automatic,
- *ECHR-DB* is augmented with rich metadata that allow to understand and use its content more easily.

The database is available at `https://echr-opendata.eu` under the **Open Database Licence (ODbL)**. The creation scripts and website sources are provided under **MIT Licence** and they are available on GitHub [19].

3.1 Database Description

From the *HUDOC* database and judgment files, we extracted, cleaned, and normalized data including descriptive and textual features. The data are available either in a structured or unstructured format:

- The unstructured format is a JSON file containing a list of all the information available about each case, including a tree-based representation of the judgment document (cf., Section 4).
- Structured information files are provided to be directly readable by popular data manipulation libraries, such as *panda* or *numpy*. Thus, they are easy to use with machine learning libraries such as *scikit-learn*. These fields include the description of cases in a flat JSON and the adjacency matrix for some important variables.

4 Database Creation Process

In this section, we outline the process of populating the *ECHR-DB* repository. The process of ingesting data is broken down into the following five tasks discussed in this section: (1) retrieving basic metadata and judgment documents, (2) cleaning cases, (3) pre-processing documents, (4) normalizing documents, and (5) generating the repository.

4.1 Retrieving Basic Metadata and Judgment Documents

Using web scrapping, basic metadata about all entries are retrieved from *HUDOC* and saved in JSON files. Common metadata include among others: case name, the language used, the conclusion in natural language. When available, we also retrieved the judgments in Microsfot Word format.

4.2 Cleaning Cases

HUDOC includes cases in various languages, cases without judgments, cases without or with vague conclusions. For this reason, its content needs to be cleaned before making it available within our project. To clean the content of *HUDOC* we applied a standard extract-transform-load (ETL) process [3]. To ensure a high quality and usability of the datasets, we cleaned and filtered out the cases. As a consequence, *ECHR-DB* includes: (1) only cases in English, (2) only cases accompanied by a judgment document, and (3) only cases with a clear conclusion, i.e., containing at least one occurrence of *violation* or *no violation.*

As part of the ETL process, we also parsed and formatted some raw data: parties are extracted from a case title and many raw strings are broken down into lists. In particular, a string listing articles discussed in a case are transformed into a list and a conclusion string is transformed into a slightly more complex JSON object. For instance, string *Violation of Art. 6-1; No violation of P1-1; Pecuniary damage - claim dismissed; Non-pecuniary damage - financial award* becomes the following list of elements:

```
{"conclusion":
  [
       {   "article": "6",
           "element": "Violation of Art. 6-1",
           "type": "violation"
       },
       {   "article": "p1",
           "element": "No violation of P1-1",
           "type": "no-violation"
       },
       {   "element": "Pecuniary damage - claim dismissed",
           "type": "other"
       },
       {   "element": "Non-pecuniary damage - financial award",
           "type": "other"
       }
   ]
}
```

In general, each item in the conclusion can have the following elements: (1) *article*: a number of the concerned article if applicable, (2) *details*: a list of additional information (paragraph or aspect of the article), (3) *element*: a part of a raw string describing the item, (4) *mentions*: diverse mentions (quantifier, e.g., 'moderate', country...), (5) *type*: of value *violation*, *no violation*, or *other.*

4.3 Pre-processing Documents

The pre-processing step consists in parsing an MS Word document to extract additional information and create a tree structure of a judgment file. During this process, we extend the set of features of a legal document with field *decision_body* with the list of persons involved in a decision, including their roles. The most important extension of a case description is the tree representation of the whole judgment document, under the field *content*. The content is described in an ordered list where each element has two fields: (1) *content* to describe the element (paragraph text or title) and (2) *elements* that represents a list of sub-elements. This tree representation eases the identification of some specific sections or paragraphs (e.g., facts or conclusion) or explore judgments with a lower granularity.

Each judgment has the same structure, which includes the following properties: (1) *Procedure*, (2) *Facts*, (3) *Law*, further composed of *Circumstances of the Case* and *Relevant Law*, and (4) *Operative Provision*.

It has been shown in [2] and [15] that each section has a different predictive power. The representation we propose allows to go further to identify each individual paragraph. Each paragraph is an independent statement (e.g., one fact for the *Facts* section, one legal argument for the *Law* section).

4.4 Normalizing Documents

In this task, judgment documents (without the conclusion) are normalized as follows: (1) tokenization, (2) stopwords removal, (3) part-of-speech tagging followed by a lemmatization, and (4) n-gram generation for $n \in \{1, 2, 3, 4\}$.

To construct a dictionary of tokens, we use *Gensim* (an open-source library for unsupervised topic modeling and natural language processing) [20]. The dictionary includes the 5000 most common tokens, based on the normalized documents. The number of tokens to use in the dictionary is a parameter of the script. The judgment documents are thus represented as a Bag-of-Words and TD-IDF matrices on top of the tree representation.

To ease data exploration, notably the connections between cases, we generated adjacency matrices for the following variables: decision body, extracted application, representatives and Strasbourg case law citations.

5 Experiments: Binary Classification

In this section, we perform a first campaign of experiments on *ECHR-DB*. Their goals are twofold. First, to studying the predictability offered by the database. Second, to provide a first baseline by testing the most popular machine learning algorithms for classification. In particular, in this paper we have focused the experiments on **determining if a specific article has been violated** or not, which is an instance of the the **binary** classification problem.

Furthermore, in this experimental evaluation, we are interested in **answering the following four questions**: (1) what is the predictive power of the data in *ECHR-DB*, (2) are all the articles equal w.r.t. predictability, (3) are some methods performing significantly better than others, and (4) are all data types (textual or descriptive) equal w.r.t. predictability?

All the experiments are implemented using *Scikit-Learn* [16]. All the experiments and scripts to analyze the results as well as to generate the plots and tables are open-source and are available on a separated GitHub repository [19] for repeatability and reusability.

5.1 Data Preparation

From *ECHR-DB* we created 11 datasets for the *binary* classification problem mentioned above. Each dataset comes in different flavors, based on: descriptive features, bag-of-words representations. These different representations (listed below) allow to study the respective importance of descriptive and textual features in the predictive models build upon the datasets:

1. *descriptive features*: structured features retrieved from HUDOC or deduced from the judgment document,
2. *bag-of-words* (BoW) representation: based on the top 5000 tokens (normalized n-grams for $n \in \{1, 2, 3, 4\}$),
3. *descriptive features + bag-of-words*: combination of both sets of features.

Each of the 11 datasets corresponds to a specific article. We kept only the articles such that there are at least 100 cases with a clear output, without consideration on the prevalence. Notice that the same case can appear in two datasets if it has in its conclusion two elements about a different article. A label corresponds to a violation or no violation of a specific article. The final datasets have been hot-one encoded. A basic description of these datasets is given in Table 1.

Table 1. Datasets description for binary classification.

	# cases	min #features	max #features	avg #features	prevalence
Article 1	951	131	2834	1183.47	0.93
Article 2	1124	44	3501	2103.45	0.90
Article 3	2573	160	3871	1490.75	0.89
Article 5	2292	200	3656	1479.60	0.91
Article 6	6891	46	3168	1117.66	0.89
Article 8	1289	179	3685	1466.52	0.73
Article 10	560	49	3440	1657.22	0.75
Article 11	213	293	3758	1607.96	0.85
Article 13	1090	44	2908	1309.33	0.91
Article 34	136	490	3168	1726.78	0.64
Article p1	1301	266	2692	1187.96	0.86

Columns min, max, and avg #features indicate the minimal, maximal, and average number of features, respectively, in the dataset cases for the representation *descriptive features + bag-of-words*.

The Bag-of-Words is a rather naive representation that loses a tremendous amount of information. However, we justify this choice by two reasons. Fist, so far, the studies on predicting the violation of articles for the ECHR cases use only a BoW representation. To be able to compare the interest of the proposed data with the previous studies, we need to use the same semantic representation. Second, from a scientific point of view, it is important to provide baseline results using the most common and established methods in order to be able to quantify the gain of more advanced techniques. This said, future work will consist in investigating advanced embedding techniques that are context aware such as LSTM or BERT-like networks. In particular, we hope not only to improve the prediction accuracy by a richer semantic, but also being able to justify a decision in natural language.

5.2 Protocol

We compared 13 standard classification methods: AdaBoost with Decision Tree, Bagging with Decision Tree, Naive Bayes (Bernoulli and Multinomial), Decision Tree, Ensemble Extra Tree, Extra Tree, Gradient Boosting, K-Neighbors, SVM (Linear SVC, RBF SVC), Neural Network (Multilayer Perceptron), and Random Forest.

For each article, we used three following flavors: (1) descriptive features only, (2) bag-of-words only, and (3) descriptive features combined with bag-of-words. For each method, each article, and each flavor, we performed a 10-fold cross-validation with stratified sample, for a total of 429 validation procedures. Due to this important amount of experimental settings, we discarded the TF-IDF representation. For the same reason, we did not perform any hyperparameter tuning at this stage.

To evaluate the performances, we reported some standard performance indicators: accuracy, F_1-score and Matthews correlation coefficient (MCC). Additionally, we report the learning curves to study the limit of the model space. The learning curves are obtained by plotting the accuracy depending on the training set size, for both the training and the test sets. The learning curves help to understand if a model underfits or overfits and thus, shape future axis of improvements to build better classifiers.

To find out what type of features are the most important w.r.t. predictability, we used a Wilcoxon signed-rank test at 5% to compare the accuracy obtained on bag-of-words representation to the one obtained on the bag-of-words combined with the descriptive features. Wilcoxon signed-rank test is a non-parametric paired difference test. Given two paired sampled, the null hypothesis assumes the difference between the pairs follows a symmetric distribution around zero. The test is used to determine if the changes in the accuracy is significant when the descriptive features are added to the textual features.

5.3 Results

Table 2 shows the best accuracy obtained for each article as well as the method and the flavor of the dataset. For all articles, the best accuracy obtained is higher than the prevalence. Linear SVC offers the best results on 4, out of 11 articles. Gradient Boosting accounts for 3, out 11 articles and Ensemble Extra Tree accounts for 2 articles.

Table 2. The best accuracy obtained for each article.

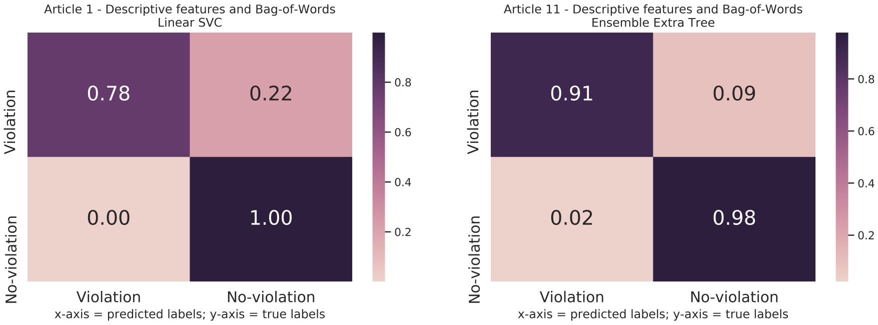
Article	Accuracy	Method	Flavor
Article 1	0.9832 (0.01)	Linear SVC	Descriptive features and Bag-of-Words
Article 2	0.9760 (0.02)	Linear SVC	Descriptive features and Bag-of-Words
Article 3	0.9588 (0.01)	BaggingClassifier	Descriptive features and Bag-of-Words
Article 5	0.9651 (0.01)	Gradient Boosting	Descriptive features and Bag-of-Words
Article 6	0.9721 (0.01)	Linear SVC	Descriptive features and Bag-of-Words
Article 8	0.9542 (0.03)	Gradient Boosting	Descriptive features and Bag-of-Words
Article 10	0.9392 (0.04)	Ensemble Extra Tree	Bag-of-Words only
Article 11	0.9671 (0.03)	Ensemble Extra Tree	Descriptive features and Bag-of-Words
Article 13	0.9450 (0.02)	Linear SVC	Descriptive features only
Article 34	0.7586 (0.09)	AdaBoost	Descriptive features only
Article p1	0.9685 (0.02)	Gradient Boosting	Descriptive features and Bag-of-Words
Average	0.9443		
Micro average	0.9644		

The standard deviation is rather low and ranges from 1% up to 4%, at the exception of article 34, for which it is equal to 9%. This indicates a low variance for the best models. The accuracy ranges from 75.86% to 98.32%, with the average of 94.43%. The micro-average that ponders each result by the dataset size is 96.44%. In general, the datasets with higher accuracy are larger and more imbalanced. For the datasets being highly imbalanced, with a prevalence from 0.64 to 0.93, other metrics may be more suitable to appreciate the quality of the results. In particular, the micro-average could simply be higher due to the class imbalance rather than the availability of data.

Regarding the flavor, 8 out 10 best results are obtained on descriptive features combined to bag-of-words. Bag-of-words only is the best flavor for article 10, whereas descriptive features - only for article 13 and article 34. This seems to indicate that combining information from different sources improves the overall results.

Figure 1 displays the normalized confusion matrix for the best methods on article 1 and 13. Similar results are observed for all the other articles. The normalization is done per line and allows to quickly figure out how the true predictions are balanced for both classes. As expected due to the prevalence, true negatives are extremely high, ranging from 0.82 to 1.00, with an average of 97.18. On the contrary, the true positive rate is lower, ranging from 0.47 to 0.91. For most articles, the true positive rate is higher than 80% and it is lower than 50% only for article 34. This indicates that the algorithms are capable of **producing models that are fairly balanced despite the fact that the classes are highly imbalanced**.

Fig. 1. Normalized confusion matrices for the best methods from Table 2.



Additionally, we provide the Matthew Correlation Coefficient (MCC) in Table 3. The MCC is generally superior to the accuracy because it takes into account the class prevalence. Therefore, it is a much better metric to estimate the model quality. The MCC ranges from 0.4918 - on article 34 to 0.8829 - on article 10. The best score is not obtained by the same article as for the accuracy (article 10 achieved 93% accuracy, below the average). Interestingly, the MCC reveals that the performances on article 34 are rather poor in comparison to the other articles and close to the performance on article 13. Surprisingly, the best method is not Linear SVC anymore (best on 3 articles) but Gradient Boosting (best on 4 articles). While the descriptive features were returning the best results for two articles, according the MCC, it reaches the best score only for article 34.

Table 3. Best Matthews Correlation Coefficient obtained for each article. The flavor and method achieving the best score for both metrics are similar for every article.

Article	MCC	Method	Flavor
Article 1	0.8654	Linear SVC	Descriptive features and Bag-of-Words
Article 2	0.8609	Linear SVC	Descriptive features and Bag-of-Words
Article 3	0.7714	BaggingClassifier	Descriptive features and Bag-of-Words
Article 5	0.7824	Gradient Boosting	Descriptive features and Bag-of-Words
Article 6	0.8488	Linear SVC	Descriptive features and Bag-of-Words
Article 8	0.8829	Gradient Boosting	Descriptive features and Bag-of-Words
Article 10	0.8411	Gradient Boosting	Bag-of-Words only
Article 11	0.8801	Ensemble Extra Tree	Descriptive features and Bag-of-Words
Article 13	0.5770	Ensemble Extra Tree	Bag-of-Words only
Article 34	0.4918	AdaBoost	Descriptive features only
Article p1	0.8656	Gradient Boosting	Descriptive features and Bag-of-Words
Average	0.7879		
Micro average	0.8163		

Once again, the micro-average is higher than the macro-average. As the MCC takes into account class imbalance, it supports the idea that adding more cases to the training set could still improve the result of these classifiers. This will be confirmed by looking at the learning curves.

Table 4 ranks the methods according to the average accuracy performed on all articles. For each article and method, we kept only the best accuracy among the three dataset flavors.

Table 4. Overall ranking of methods according to the average accuracy obtained for every article.

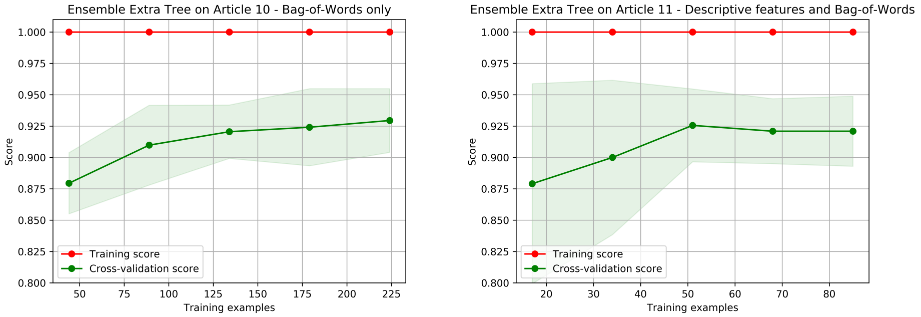
Method	Accuracy	Micro Accuracy	Rank
Ensemble Extra Tree	0.9420	0.9627	1
Linear SVC	0.9390	0.9618	2
Random Forest	0.9376	0.9618	3
BaggingClassifier	0.9319	0.9599	4
Gradient Boosting	0.9309	0.9609	5
AdaBoost	0.9284	0.9488	6
Neural Net	0.9273	0.9535	7
Decision Tree	0.9181	0.9419	8
Extra Tree	0.8995	0.9275	9
Multinomial Naive Bayes	0.8743	0.8907	10
Bernoulli Naive Bayes	0.8734	0.8891	11
K-Neighbors	0.8670	0.8997	12
RBF SVC	0.8419	0.8778	13
Average	0.9086	0.9335	

Surprisingly, neither Linear SVC nor Gradient Boosting are the best methods with a respective rank of 2 and 5, but the best one is Ensemble Extra Tree. Random Forest and Bagging with Decision Tree are the second and third ones, respectively, and they never achieved the best result on any article. It simply indicates that these methods are more consistent across the datasets than Linear SVC and Gradient Boosting.

Figure 2 displays the learning curves obtained for the best methods on articles 10 and 11. The training error becomes (near) zero on every instance after only few cases, except for article 13 and 34. The test error converges rather fast and remains relatively far from the training error, synonym of high bias. Those two elements indicate underfitting. Similar results are observed for all methods. Usually, more training examples would help, but since the datasets are exhaustive w.r.t. the European Court of Human Rights cases, this is not possible. As a consequence, we recommend using a more complex model space and hyperparameter tuning. In particular, as mentioned above, the usage of more advanced embedding techniques is an obvious way to explore. Finally, an exploratory analysis of the datasets may also help in removing some noise and finding the best predictors.

If we assume that the process of deciding if there is a violation or not is the same, independently of the article, a solution might be a transfer learning, to leverage what is learnable from the other articles. We let this research axis for future work.

Fig. 2. Learning curves for the best methods as described by Table 2.



Finally, we used a Wilcoxon signed-rank test at 5% to compare the accuracy obtained on the bag-of-words representation to the one obtained on the bag-of-Words combined with the descriptive features. The difference between the samples has been found to be significant only for article 6 and article 8. The best result obtained on bag-of-Words is improved by adding descriptive features for every article. However, statistically, for a given method, adding descriptive features does not improve the result. Additionally, we performed the test per method. The result is significant for any method.

In conclusion, the datasets demonstrated a strong predictability power. Apart from article 13 and 34, each article seems to provide similar results, independently of the relatively different prevalence. If the accuracy is rather high, a more informative metric, such as MCC, shows that there are still margins of improvements. Hyperparameter tuning [18] is an obvious way to go, and this preliminary work has shown that good candidates for fine tuning are Ensemble Extra Tree, Linear SVC, and Gradient Boosting.

5.4 Discussion

To sum up, we achieved an average accuracy of 94% which is respectively 15pp and 19pp higher than [2] and [15]. The size of the dataset does matter since we showed that the model underfits. Also, we showed that SVM is far from being the best method for all articles. However, such a huge gap cannot be explained only by those two factors.

In our opinion, the main problem with the previous studies is that the authors rebalanced their datasets. As those datasets were highly imbalanced, they used undersampling, which resulted in a very small training dataset. Most likely, the training dataset was not representative enough of the feature space which leads to underfitting (even more than in our experiments). They justified that rebalancing was necessary to ensure that the classifier was not biased towards a certain class. For this reason, we argue that they modified the label distribution. As some classification methods rely on the label distribution to learn, they introduce themselves a prior shift [13]. In general, rebalancing is necessary only when, indeed, the estimator is badly biased. It is true that the accuracy is meaningless on imbalanced datasets but we can still control the quality of the model using a collection of more robust indicators, including among others: F1-score, MCC, and normalized confusion matrices. In other words, our approach (discussed in this paper) is more neutral in the sense we do not change the label distribution, and it still offers a robust classifier.

This experimental campaign has demonstrated that the textual information provides better results than descriptive features alone, but the addition of the descriptive feature improved in general the result of the **best** method. We emphasize the best method (obtained among all methods) because for a given method adding the descriptive feature are not significantly improving the results.

Another way of improving the results is to tune the different phases of the dataset generations. In particular, our preliminary work reported in [18] has shown that 5000 tokens and 4-grams might not be enough to take the best out of the documents. It might seem surprising, but the justice language is codified and standardized in a way that n-grams for large n might contain better predictors for the outcome.

6 Conclusion

In this paper, we presented an open repository, called *ECHR-DB*, of legal cases and judicial decision justifications. The main purposes of constructing the repository are as follows. First, to provide cleaned and transformed content from the repository of the European Court of Human Rights, that is ready to use by researchers and practitioners. Second, to augment original legal documents with metadata, which will ease the process of analyzing these documents. Third, to provide a benchmark with baseline results for classification models in the legal domain, for other researchers.

Currently, *ECHR-DB* is the largest and most exhaustive repository of legal documents from the European Court of Human Rights. It includes several types of data that can be easily used to reproduce various experiments that have been done so far by other researchers. We argue that providing the final data is not enough to ensure quality and trust. In addition, there are always some opinionated choices in the representation, such as the number of tokens, the value of n for the n-grams calculation or the weighting schema in the TF-IDF transformation. As a remedy, we provide the whole process of dataset construction from scratch. The process is implemented by means of Python scripts and available on GitHub [4].

The experiments on *ECHR-DB* provide a baseline for future work on classification. The predictability power of each dataset has been tested for the most popular machine learning methods. We achieved the average accuracy of 0.9443. The learning curves have shown that the models are underfitting but, as the datasets are exhaustive, it is not possible to provide more examples. We showed that the textual features help in determining the outcome. Combining descriptive and textual features always help for the best classifier, but overall, the results are not better statistically. Descriptive features surprisingly hold reasonable predictive power.

The preliminary experiments provide several axes of improvements, e.g., better embedding with state of the art encoders, hyperparameter tuning, multi-stage classifier, and transfer learning. From the results, it seems clear that predicting if an article has been violated or not can be handled with the current state of the art in artificial intelligence. However, many interesting questions and problems arise from the proposed repository, e.g. *can we provide legal justification in natural language to a prediction?*, which will be addressed in the future work.

References

1. Maastricht University Law and Tech Lab, `https://www.maastrichtuniversity.nl/about-um/faculties/law/research/law-and-tech-lab`
2. Aletras, N., Tsarapatsanis, D., Preoiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the European Court of Human Rights: a natural language processing perspective. PeerJ Computer Science **2**, e93 (2016)
3. Ali, S.M.F., Wrembel, R.: From conceptual design to performance optimization of ETL workflows: current state of research and open problems. VLDB Journal **26**(6), 777–801 (2017)
4. Ashley, K.D.: Artificial intelligence and legal analytics: new tools for law practice in the digital age. Cambridge University Press (2017)
5. Atkinson, K., Bench-Capon, T.: Reasoning with legal cases: Analogy or rule application? In: Proc. of Int. Conf. on Artificial Intelligence and Law (ICAIL). pp. 12–21. ACM (2019)
6. Bilalli, B., Abelló, A., Aluja-Banet, T., Wrembel, R.: Intelligent assistance for data pre-processing. Computer Standards & Interfaces **57**, 101–109 (2018), `https://doi.org/10.1016/j.csi.2017.05.004`
7. Bilalli, B., Abelló, A., Aluja-Banet, T., Wrembel, R.: PRESISTANT: learning based assistant for data pre-processing. Data Knowl. Eng. **123** (2019), `https://doi.org/10.1016/j.datak.2019.101727`
8. Crone, S.F., Lessmann, S., Stahlbock, R.: The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing. Eur. J. Oper. Res. **173**(3), 781–800 (2006)
9. Dasu, T., Johnson, T.: Exploratory data mining and data cleaning, vol. 479. John Wiley & Sons (2003)

[4] `https://github.com/aquemy/ECHR-OD_predictions`

10. Guimerà, R., Sales-Pardo, M.: Justice Blocks and Predictability of U.S. Supreme Court Votes. PLoS ONE **6**(11), e27188 (2011)
11. Katz, D.M., Bommarito, M.J., Blackman, J.: A general approach for predicting the behavior of the Supreme Court of the United States. PLOS ONE **12**(4), e0174698 (2017)
12. Kelleher, J.D., Mac Namee, B., D'Arcy, A.: Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies. MIT Press (2015)
13. Lemberger, P., Panico, I.: A primer on domain adaptation (2020)
14. Martin, A.D., Quinn, K.M., Ruger, T.W., Kim, P.T.: Competing approaches to predicting supreme court decision making. Perspectives on Politics **2**(4), 761–767 (2004)
15. Medvedeva, M., Vols, M., Wieling, M.: Using machine learning to predict decisions of the European Court of Human Rights. Artifficial Intelligence and Law (2019), `https://doi.org/10.1007/s10506-019-09255-y`
16. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
17. Quemy, A.: Data science techniques for law and justice: Current state of research and open problems. In: Proc. of ADBIS Workshops. pp. 302–312. Springer, CCIS 767 (2017)
18. Quemy, A.: Data pipeline selection and optimization. In: Proc. Int. Worksh. on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP) (2019)
19. Quemy, A.: Predictions of the European Court of Human Rights (2019), `https://github.com/aquemy/ECHR-OD_predictions`
20. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proc. of Worksh. on New Challenges for NLP Frameworks. pp. 45–50. ELRA (2010)
21. Rissland, E.L.: AI and Similarity. IEEE Intelligent Systems **21**(3), 39–49 (2006)
22. Ruger, T.W., Kim, P.T., Martin, A.D., Quinn, K.M.: The supreme court forecasting project: Legal and political science approaches to predicting supreme court decisionmaking. Columbia Law Review **104**(4), 1150–1210 (2004)
23. Yan, L., Wilson, C.: Developing AI for law enforcement in Singapore and Australia. Commun. ACM **63**(4) (2020)