Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

ECHR-DB: On building an integrated open repository of legal documents for machine learning applications

Alexandre Quemy [a,b,*], Robert Wrembel [a]

[a] IBM Krakow Software Lab, Cracow, Poland
[b] Poznan University of Technology, Poznań, Poland

ARTICLE INFO

Article history:
Received 14 October 2020
Received in revised form 25 March 2021
Accepted 6 April 2021
Available online xxxx
Recommended by Andrea Tagarelli

Keywords:
Open data repository
Legal documents repository
Judgment documents
European Court of Human Rights
Machine learning
Classification of legal documents

ABSTRACT

This paper presents an exhaustive and unified repository of judgments documents, called *ECHR-DB*, based on the European Court of Human Rights. The need of such a repository is explained through the prism of the researcher, the data scientist, the citizen, and the legal practitioner. Contrarily to many open data repositories, the full creation process of *ECHR-DB*, from the collection of raw data to the feature transformation, is provided by means 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 one of the most important issues in data governance nowadays. The experimental evaluation was performed to study the problem of predicting the outcome of a case, and to establish baseline results of popular machine learning algorithms. The obtained results are consistently good across the binary datasets with an accuracy comprised between 75.86% and 98.32%, having the average accuracy equals to 96.45%, which is 14pp higher than the best known result with similar methods. We achieved a F1-Score of 82% which is aligned with the recent result using BERT. We show that in a multilabel setting, the features available prior to a judgment are good predictors of the outcome, opening the road to practical applications.

We observe a shift from reasoning techniques and expert systems to data-centric approaches [1,2], which use ML algorithms. However, to provide satisfactory prediction ML models need to be trained on large datasets. The availability of such real datasets is limited in practice, and it remains a major problem for researchers and practitioners. There exist few initiatives to provide unified repositories of clean data. Moreover, they are limited to specific courts, rather incomplete or behind paywalls. These observations motivated us to build an open repository of judgment documents. For the European Court of Human Rights, there exists database HUDOC[1] that contains all judgments since its creation. However, it is impossible to access multiple documents at once and case documents are not unified in the way that they offer data in a tabular format and free unstructured texts. There exists two significant databases [3,4]. However, they are both static in the sense they are not updated by the authors. On the contrary, the database proposed in this paper is **automatically updated every month** with new cases. Additionally, [3,4] provide only textual information, and no meta-data or additional information available about each case. Finally, [4] is not easily available since it requires contacting the authors to access the data. In other words, despite their public availability, it is difficult to access the data and work with them.

For these reasons, the overall **goal** of this project is to provide an exhaustive and unified set of data, along with metadata, about one of the main European legal institution, namely the European Court of Human Rights. The importance of such work is as follows:

- to draw the attention of researchers on this domain that has important consequences on the society;
- to allow researchers and practitioners to easily study the European Court of Human Rights;
- to provide a unified benchmark to compare ML techniques dedicated to the legal domain;
- to provide a similar and more complete repository for Europe as it already exists for the United States judicial system, notably because the law systems are different in both sides of the Atlantic.

The **contributions** of this paper can be summarized as follows.

- First, we provide a benchmark set for ML algorithms. It is composed of (almost) all cases judged by the European Court of Human Rights since its creation. The data is cleaned and transformed to ease the exploration and usage of ML algorithms.

* Corresponding author at: IBM Krakow Software Lab, Cracow, Poland.
E-mail addresses: aquemy@pl.ibm.com (A. Quemy), robert.wrembel@cs.put.poznan.pl (R. Wrembel).

[1] https://hudoc.echr.coe.int.

https://doi.org/10.1016/j.is.2021.101822
0306-4379/© 2021 Elsevier Ltd. All rights reserved.

- Second, we provide the whole data extraction, transformation, integration, and loading (ETL) pipeline used to generate the benchmark data repository, as the open-source software. This technical contribution aims at increasing the trust in the processed data and ease future iterations of the benchmark, to integrate new cases and data per case.
- Third, we provide exhaustive and high-quality repository, called *ECHR-DB*, of judicial documents for diverse ML problems in the legal domain, based on the European Court of Human Rights documents.
- Fourth, we present the first analysis of the benchmark with standard classification algorithms, in order to predict the outcome of cases and establish baseline models for comparison with future studies. The experiments study binary classification, like previous studies, but also multiclass and multilabel classification, which represent more complex situations.

This paper comes with *supplementary material* available on GitHub.[2] It contains additional examples about the data format, as well as all secondary results of the experiments that we omitted due to space constraints.

The plan of this paper is as follows. The whole ETL pipeline is presented in details in Section 3. Section 4 presents the datasets for our experiments. Sections 5, 6, and 7 discuss the results of experiments on the quality of prediction models built by binary, multiclass, and multilabel classification algorithms, respectively. Finally, Section 8 concludes the paper by discussing the remaining challenges and future work. Appendix A outlines the functionality of *ECHR-DB* and its user interface.

This paper extends our preliminary work [5]. Among other extensions, a normalized SQL has been created to allow more complex data manipulation than on tabular data. We performed a multiclass and multilabel classification experiments which are more complex problems than the original binary classification one. We also provide additional results for the binary classification experiments in order to better support our conclusions.

1. Related work

Predicting the outcome of a justice case is challenging, even for the best legal experts. As shown in [6], 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*[3] project reached 85.20% and 84.85% correct predictions, respectively.

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[4] of the *Supreme Court of the United States* and the *HUDOC* database[5] 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 is hard to retrieve and to work with. As mentioned earlier, there are two noticeable efforts to provide ECHR database [3,4], but both database are not exhaustive, does not provide metadata and are not regularly updated.

The prediction of the *Supreme Court of the United States* has been widely studied, notably through the *SCOTUS* repository [7–9]. To the best of our knowledge, the only predictive models that used the content of *HUDOC* were reported in [10,11] and [3]. The data used in [10] 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 [11] to reach an overall of 75% accuracy on judgment documents up to September 2017. Finally, in [3] several attention-based neural networks and BERT [12] are compared and achieved a F1-score of 82%. Additionally, they achieved a F1-score of 60.8% in a multilabel setting in which the presence of label denotes a violation.

New studies tend to suggest that there will always be a limit in reasoning systems to handle new cases presenting novel situations [13], 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*).

On the contrary, [14] argues that because of the specificities of the legal domain such as gray areas of interpretation, non-monotonicity in reasoning or many exceptions, Machine Learning methods still do not perform as spectacularly as they do in many other fields. Therefore, the author explains that traditional AI is still needed in combination with data-centric method. For a survey of legal analytics approaches, we refer the reader to [15].

Another central problem in legal analytics is the Justification problem which consists in explaining a given judicial decision. The problem is different from Explainable AI as noted in [15] because a model might take decisions based on non-legal factors while a legal justification, by definition, needs to fit into the scope of a legal framework. We believe that our repository provides useful formats to tackle this issue. In particular, the tree representation and the detailed conclusion elements can be used to cite arguments in a similar fashion as CBR system such as CATO [16] or to automatically build Abstract Argumentation Frameworks [17–19]. Also, by keeping the well-defined structure of the judgment documents, we hope it will allow for better corpus embeddings representation than the Bag-of-Words or even more advanced Deep-Learning based techniques such as LSTM [20] or BERT [12] that are unaware of this structure.

2. ECHR-DB overview

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 Europe as it already exists in the United States, notably because the law systems are different in both sides of the Atlantic.

The final data is available for direct download through a portal available at https://echr-opendata.eu under the **Open Database License (ODbL)**.

The creation scripts and website sources are provided under **MIT License** and they are available on GitHub [21]. The project offers data in several format:

[2] https://echr-od.github.io/ECHR-OD_project_supplementary_material/.
[3] https://fantasyscotus.lexpredict.com/.
[4] http://scdb.wustl.edu/.
[5] https://hudoc.echr.coe.int/eng.

- 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 3).
- Structured information files are provided in JSON and CSV and are meant to be directly readable by popular data manipulation libraries, such as Pandas or Numpy . Thus, they are easy to use with machine learning libraries such as Scikit-Learn. It includes the description of cases in a flat JSON and the adjacency matrix for some important variables such as the body members, the cross-references between cases or the representatives. Additionally, the ready-to-use Bag-of-Words and TF–IDF representations of judgments are also available.

We also provide a normalized SQL database for more advanced queries. Finally, the portal allows to explore online the data or to interface it with external applications through a well documented REST API. More details about its implementation and user interface is provided in Appendix A. Examples of usage and visualizations are available at https://echr-opendata.eu/charts/.

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

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

3. Data processing pipeline of ECHR-DB

In this section, we discuss a full data processing pipeline used to build the integrated repository (database) of judicial cases — *ECHR-DB*.

The processing pipeline that we have used to build *ECHR-DB* is shown in Fig. 1. The process of ingesting data is broken down into the following five steps discussed in this section, i.e.,: (1) ingesting judgment documents and basic metadata, (2) cleaning cases, (3) pre-processing documents, (4) normalizing documents, and (5) generating the repository.

3.1. Retrieving judgment documents and basic metadata

Using web scrapping, we retrieved all entries from Hudoc. The available data consist of basic metadata and the judgment document in natural language. Metadata include: case name, the application number, the language used, the conclusion in a natural language, plaintiff representative, the parties, the decision body members and their role, the other cases cited in a given case, the Strasbourg Case Law mentioned in a given case, the decision date, the introduction date, the doctype branch, the external sources cites in a given case (e.g. national laws), the importance of the case, the originating body, the respondent, if there are separate opinion for a given case, the articles a given case is about (which are defined during the application). We also retrieved the judgments documents in Microsoft Word format. MS Word is a proprietary format. However, its structure in XML is easier to parse than a PDF which is the reason why we used it.

3.2. Cleaning metadata

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 for further processing. To clean the content of Hudoc we applied a standard extract–transform–load (ETL) process [22]. 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 articles, if applicable, (2) *details*: a list of additional information (a 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’, ‘substantial aspect’ or ‘conditional’, geographic precision, e.g., ‘Kyrgyzstan’ for a case filed against Russia for an extradition to Kyrgyzstan…), (5) *type*: of value *violation*, *no violation*, or *other*.

To ensure a high quality and usability of the data, the process cleaned and filtered out the cases. As a consequence, *ECHR-DB* includes: (1) only cases in English, and (2) only cases with a clear conclusion, i.e., containing at least one occurrence of *violation* or *no violation*. Therefore, we removed cases with conclusion ‘Struck out of the list’, ‘Revision rejected’ or ‘Inadmissible’ which represent 1847 cases. Additionally, three cases has been removed because of a broken judgment file that could not be parsed.

Finally, on top of saving the case information in a JSON file, we output a JSON file for each unique article with at least 100 associated cases.[6] Additionally, some basic statistics about the attributes are generated, e.g. the cardinality of the domain and the density (i.e. the cardinality over the total number of cases). For instance, the attribute `itemid` is unique and thus, as expected, its density is 1.

In comparison, the field `article_` (raw string containing a list of articles discussed in a case – not kept in the final database –) and `article` (its parsed and formatted counterpart) have a density of respectively 0.26 and 0.01. This illustrates the interest of our processing method: using the raw string, the article attribute is far more unique than it should be. In reality, there are about

[6] This constant is a parameter of the script and can thus be modified for additional experimentations.

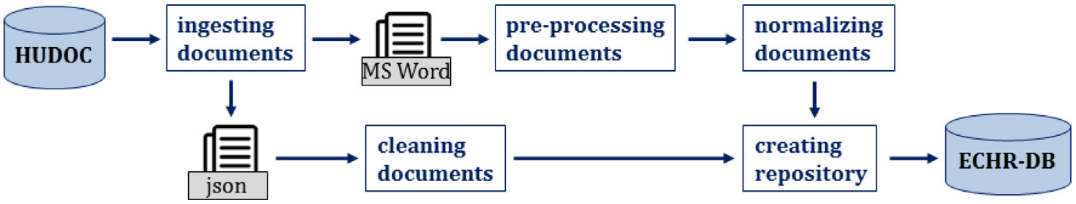


Fig. 1. The processing pipeline for building *ECHR*.

130 different values that are really used across the datasets and that represents all the possible combinations of articles discussed accross cases.

We denote by *descriptive features* the data and metadata that are not the judgment document.

3.3. Pre-processing judgment documents

The pre-processing task consists in parsing judgment documents in MS Word format to extract additional information and create a tree structure of a judgment file. For each case, the metadata is extended with some additional information such as the decision body, i.e. 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* that describes 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.

```
{
  "content":{
    "001-155097.docx":[
      {
        "content":"PROCEDURE",
        "elements":[
          {
          "content":"1. The case originated in an application [...].",
            "elements":[
            ]
          },
          "..."
        ]
      },
      {
        "content":"THE FACTS",
        "elements":[
          "content": "I. THE CIRCUMSTANCES OF THE CASE",
          "elements": [
            "..."
          ]
        ]
      },
      "...",
      {
        "content": "FOR THESE REASONS, THE COURT, UNANIMOUSLY,",
        "elements":[
          "..."
        ],
        "section_name": "conclusion"
      }
    ]
  }
}
```

Each judgment has the same structure, which includes the following sections: (1) *Procedure*,(2) *Law* and (3) *Facts*, that is further composed of: *Circumstances of the Case* and *Relevant Law*, as well as (4) *Operative Provisions*. In [11] and [10] it has been shown that each section has a different predictive power. The representation that we propose allows to go further to identify each individual paragraph.

3.4. Normalizing documents

In this task, judgment documents (without the conclusion) are normalized by means of: (1) part-of-speech tagging, (2) tokenization, (3) stopwords removal, followed by a lemmatization, and (4) n-gram generation for $n \in \{1, 2, 3, 4\}$. The list of stopwords is provided by NLTK [23].

Part-of-speech tagging. This step consists in associated words with their category depending on the context. For instance, “fire” can be either a noun or a verb depending on the sentence. This is particularly useful to extract entities but also to properly clean the documents from words that carry no semantic.

Tokenization. This step consists in breaking down sentences into words. It is greatly helped by the part-of-speech tagging.

Stopwords removal. Once the part-of-speech and tokenization is done, we removed words that carry no semantic information (stopwords) such as “a”, “the”, etc.

***n*-gram generation**. A n-gram is a contiguous sequence of n words from a text. For instance, the sequence “new york city” provides three unigram, two 2-grams (“new york” and “york city”) and one single 3-gram. For probabilistic models which cover most machine learning algorithms, n-grams are important to statistically address such ambiguous statements. The true semantic of “new york city” is lost when considering only the three separate words. In addition, the 2-grams cannot distinguish between the city of New-York and the state of New-York.

To construct the final dictionary of tokens, we use an open-source library for unsupervised topic modeling and natural language processing — Gensim [24]. The dictionary includes the 5000 most common tokens, based on 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.

3.5. Creating repository

Once data and metadata have been generated, 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. Finally, using all the generated data, we created a normalized SQL database to allow for more complex queries then what is possible to achieve on JSON or CSV.

The SQL database is composed of one main table `case`. The table `case` has several n-to-n relationships: `representative`, `party`, `decisionbodymember`, `scl` (Strasbourg Case Law) and `conclusion`. The table `conclusion` is itself composed of two n-to-n relationships: `mention` and `detail`. Finally, The table `case` has many 1-to-n relationships: `article`, `issue`, `externalapp`, `documentcollectionid`, `kpthesaurus` and `externalsources`. For each case, the tree representation of the judgment document is provided in a JSON field. The relational schema of the normalized SQL database is available in Appendix B.

4. Datasets for classification

In this section, we describe the datasets extracted from *ECHR-DB* for the purpose of our experiments. The **goal of the experiments** is twofold. First, to study the predictability offered by the database. Second, to provide the additional baselines by testing the most popular machine learning algorithms for classification as previous studies used only SVM. In this paper, we have focused the experiments on determining the outcome of new cases. The problem can be seen as a classification problem: is a law article being violated or not?

In these experimental evaluations, we are interested in answering the following four questions, in particular:

- what is the predictive power of the data in *ECHR-DB*,
- are all the articles equal w.r.t. predictability,
- are some methods performing significantly better than others, and
- are all data types (textual or descriptive) equal w.r.t. predictability?

To answer these questions, we studied three variations of the classification problem, namely: binary (described in Section 5, multiclass (cf. Section 6), and multilabel classification (cf. Section 7).

The problem of classification consists in finding a mapping from an input vector space $\mathcal{X}$ to a discrete decision space $\mathcal{Y}$ (the classes) using a set of examples. It is often viewed as an approximation problem s.t. we want to find a model h of an unknown mapping f available only through a sample called *training set*. A training set $(\mathbf{X}, \mathbf{y})$ consists of N input vectors $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and their associated correct class $\mathbf{y} = \{y_i = f(\mathbf{x}_i)\}_{i=1}^{N}$.

We aim at finding h that minimizes the empirical classification error:

$$\min_h \sum_{(\mathbf{x},y)\in(\mathbf{X},\mathbf{y})} \mathbb{I}_{\{y \neq h(\mathbf{x})\}}. \tag{1}$$

The binary classification problem is a special case such that $\mathcal{Y}$ has only two elements (in our case, violation or no violation of a given article). In a multilabel classification setting, several labels can be assigned to a single element. In particular, in our case, each label consists of an article and if it is violated or not.

All the experiments are implemented using Scikit-Learn [25]. The datasets are extracted from the ECHR-DB database built in December 2020. 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 [21] for repeatability and reusability.

From *ECHR-DB*, we created 11 datasets for the *binary* classification problem, one for the multiclass problem and one for the multilabel problem. Each dataset comes in different flavors, based on the descriptive features and 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 and metadata 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 + BoW*: combination of both sets of features.

For each dataset, we removed the conclusion, as well as the following metadata: the articles discussed in the case (field `articles`) and the conclusion (field `conclusion`). We discarded the articles for which there are less than 100 cases. For binary classification, each dataset corresponds to a specific article. Notice that the same case can appear in several datasets if it has in its conclusion several elements about different articles. A *label* corresponds to a violation or no violation of a specific article. The descriptive features have been one-hot encoded for non-numeric variables. A basic description of these datasets is given in Table 1.

Table 1
Datasets description for binary classification.

	# cases	Violation	No-violation	Prevalence
Article 2	808	717	91	0.89
Article 3	2307	2044	263	0.89
Article 5	2230	2022	208	0.91
Article 6	7216	6503	713	0.90
Article 8	1363	998	365	0.73
Article 10	667	518	149	0.78
Article 11	271	231	40	0.85
Article 13	1849	1741	108	0.94
Article 14	415	210	205	0.51
Article 34	167	109	58	0.65
Article p1–1	1355	1222	133	0.90

Columns min, max, and avg #features indicate the minimal, maximal, and average number of features, respectively, in the cases for the representation *descriptive features and bag-of-words*. The column prevalence indicates the proportion of violations.

Bag-of-Words is a rather naive representation that loses a substantial amount of information. However, we justify this choice by two reasons. First, so far, the studies on predicting the violation of articles for the *ECHR* cases use only the BoW representation.[7] 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. Future work will consist of investigating advanced embedding techniques that are context aware such as LSTM or BERT-like networks in a similar fashion as in [3]. 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.

For multiclass classification, there exists 16 different classes in total (the number of different articles multiplied by two possible decisions: violation or no violation). To create the multiclass dataset, we aggregated different binary classification datasets by removing the cases present in several datasets. When a case has several article in its conclusion, we kept as label the first one given by HUDOC. A description of these datasets is given in Table 2. Notice that this multiclass setting differs from [3] as a non-violation of a given article is also a label to be identified. Also, after applying our merging method, articles 13, 14 and 34 had less than 100 cases and have been discarded.

For multilabel classification, there exists 22 different labels and the main difference with the multiclass is that there is no need to remove cases that appear in multiple binary classification datasets. The labels are simply stacked. Table 3 summarizes the dataset composition. Given the class imbalance for all datasets, the reader might wonder why they are not rebalanced. We justify this choice a posteriori by the analysis of the confusion matrices and learning curve (Section 5.2) and discuss it further in Section 5.3.

Fig. 2 shows the labels repartition among the multiclass and multilabel datasets. Fig. 3 shows the histogram of label numbers and cases per label.

[7] At the moment the experiments have been conducted, [3] was not published. However, the results we obtained with classic Machine Learning methods are better the state of the art Deep Learning methods of [3].

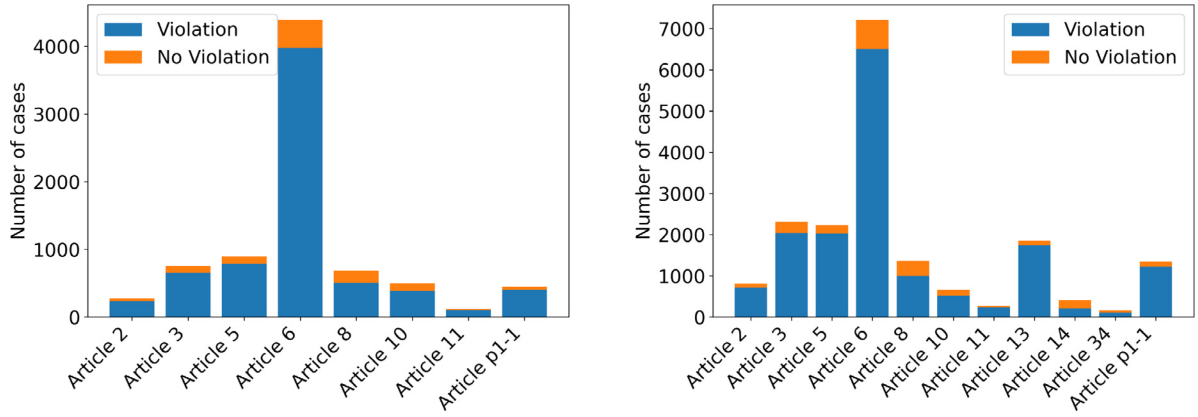


Fig. 2. The number of cases depending on the article and the outcome for the multiclass dataset (left) and multilabel dataset (right).

Table 2
Dataset description for the multiclass dataset.

	# cases	Violation	No-violation	Prevalence
Article 2	273	236 (0.033)	37 (0.037)	0.86
Article 3	756	655 (0.093)	101 (0.100)	0.87
Article 5	897	788 (0.112)	109 (0.108)	0.88
Article 6	4394	3980 (0.563)	414 (0.411)	0.91
Article 8	687	506 (0.072)	181 (0.180)	0.74
Article 10	497	389 (0.055)	108 (0.107)	0.78
Article 11	121	103 (0.015)	18 (0.018)	0.85
Article p1–1	446	407 (0.058)	39 (0.039)	0.91

For each article the following features are indicated: (1) the number of cases, (2) the number of cases labeled as violated and not violated (in parenthesis, the prevalence w.r.t. the whole dataset), and (3) the prevalence per article.

Table 3
Dataset description for the multilabel dataset.

	# cases	Violation	No-violation	Prevalence
Article 2	808	717 (0.059)	91 (0.007)	0.89
Article 3	2307	2044 (0.167)	263 (0.022)	0.89
Article 5	2230	2022 (0.165)	208 (0.017)	0.91
Article 6	7216	6503 (0.532)	713 (0.058)	0.9
Article 8	1363	998 (0.082)	365 (0.030)	0.73
Article 10	667	518 (0.042)	149 (0.012)	0.78
Article 11	271	231 (0.019)	40 (0.003)	0.85
Article 13	1849	1741 (0.142)	108 (0.009)	0.94
Article 14	415	210 (0.017)	205 (0.017)	0.51
Article 34	167	109 (0.009)	58 (0.005)	0.65
Article p1–1	1355	1222 (0.100)	133 (0.011)	0.9

For each article the following features are indicated: (1) the number of cases, (2) the number of cases labeled as violated and not violated. In parenthesis, the label prevalence w.r.t. the whole dataset.

5. Experiments: Binary classification

5.1. Protocol

We compared 12 standard classification algorithms provided by Scikit-Learn, namely: AdaBoost with Decision Tree, Bagging with Decision Tree, Naive Bayes (Bernoulli and Multinomial), Decision Tree, Ensemble Extra Tree, Extra Tree, Gradient Boosting, K-Neighbors, Linear SVM,[8] Neural Network (Multilayer Perceptron), and Random Forest.

For each article, we used the following three 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 and used the default parameters of the Scikit-Learn implementation.

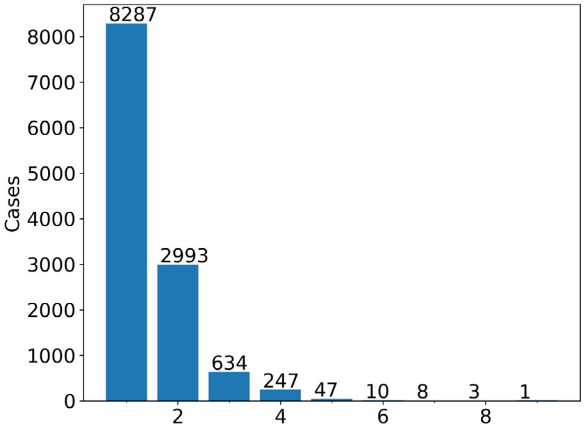


Fig. 3. The number of cases depending on the number of labels for the multilabel dataset.

To evaluate the performances, we reported some standard performance metrics, namely: (1) accuracy, (2) F_1-score, and (3) MCC. We recall the definition of these metrics. Denoting by: TP — the number of true positives, TN — the number of true negatives, FP — the number of false positives, and FN — the number of false negative. For the sake of completeness, we include the standard definitions of these metrics.

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{2}$$

$$\mathrm{F}_1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{3}$$

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{4}$$

The values of accuracy, F_1-score, and MCC are within ranges [0, 1], [0, 1] and [−1, 1], respectively (values closer to 1 are better). MCC has been shown to be more informative than other metrics derived from the confusion matrix [26], in particular with imbalanced datasets.

[8] Experiments have been conducted with Radial-Based Function SVM on a previous version of the datasets and as the method constantly gives the worst results we discarded it on the current datasets.

Table 4
The best accuracy obtained for each article. Standard deviation is reported between brackets.

Article	Accuracy	Method	Flavor
2	0.9670 (0.02)	Linear SVC	Bag-of-Words only
3	0.9489 (0.01)	Gradient Boosting	Descriptive features and Bag-of-Words
5	0.9611 (0.01)	Linear SVC	Descriptive features and Bag-of-Words
6	0.9749 (0.00)	Gradient Boosting	Descriptive features and Bag-of-Words
8	0.9574 (0.02)	Gradient Boosting	Descriptive features and Bag-of-Words
10	0.9587 (0.02)	BaggingClassifier	Descriptive features and Bag-of-Words
11	0.9520 (0.04)	Ensemble Extra Tree	Bag-of-Words only
13	0.9672 (0.01)	Ensemble Extra Tree	Descriptive features and Bag-of-Words
14	0.8994 (0.03)	Linear SVC	Bag-of-Words only
34	0.7254 (0.07)	Neural Net	Descriptive features and Bag-of-Words
p1–1	0.9751 (0.02)	Linear SVC	Descriptive features and Bag-of-Words
Average	0.9352		
Micro average	0.9628		

Legend: *desc* — descriptive features; *BoW* — bag of words.

Table 5
The best MCC obtained for each article.

Article	MCC	Method	Flavor
2	0.8569	Linear SVC	Descriptive features and Bag-of-Words
3	0.7493	Gradient Boosting	Descriptive features and Bag-of-Words
5	0.7744	Linear SVC	Descriptive features and Bag-of-Words
6	0.8486	Linear SVC	Descriptive features and Bag-of-Words
8	0.8890	Gradient Boosting	Descriptive features and Bag-of-Words
10	0.8801	BaggingClassifier	Descriptive features and Bag-of-Words
11	0.8303	Ensemble Extra Tree	Bag-of-Words only
13	0.6506	Ensemble Extra Tree	Descriptive features and Bag-of-Words
14	0.8050	Linear SVC	Bag-of-Words only
34	0.3287	Linear SVC	Descriptive features and Bag-of-Words
p1–1	0.8509	Linear SVC	Descriptive features and Bag-of-Words
Average	0.7694		
Micro average	0.8065		

The flavor and method achieving the best score for both metrics are the same for every article. Legend: *desc* — descriptive features; *BoW* — bag of words.

Table 6
The best F1-score obtained for each article.

Article	F1 score	Method	Flavor
2	0.9661	Linear SVC	Descriptive features and Bag-of-Words
3	0.9451	Gradient Boosting	Descriptive features and Bag-of-Words
5	0.9600	Linear SVC	Descriptive features and Bag-of-Words
6	0.9742	Linear SVC	Descriptive features and Bag-of-Words
8	0.9570	Gradient Boosting	Descriptive features and Bag-of-Words
10	0.9581	BaggingClassifier	Descriptive features and Bag-of-Words
11	0.9527	Ensemble Extra Tree	Bag-of-Words only
13	0.9614	Ensemble Extra Tree	Descriptive features and Bag-of-Words
14	0.8986	Linear SVC	Bag-of-Words only
34	0.6780	Linear SVC	Descriptive features only
p1–1	0.9741	Linear SVC	Descriptive features and Bag-of-Words
Average	0.9296		
Micro average	0.9608		

The flavor and method achieving the best score for both metrics are the same for every article. Legend: *desc* — descriptive features; *BoW* — bag of words.

Additionally, we report the learning curves to study the limit of the model space. The learning curves are obtained by plotting the accuracy as a function of a 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*, i.e., a non-parametric paired difference test at 5%, in order to compare the accuracy obtained on bag-of-words representation to the one obtained on the bag-of-words combined with the descriptive features. Given two paired samples, the null hypothesis assumes that the difference between the pairs follows a symmetric distribution around zero. The test is used to determine if the changes in the accuracy are significant when the descriptive features are added to the textual features.

5.2. Results

Table 4 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 articles and Ensemble Extra Tree accounts for 2 articles.

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 7%. This indicates a low variance for the best models. The accuracy ranges from 72.54% to 97.51%, with the average of 93.52%. The micro-average that ponders each result by the dataset size is 96.28%. In general, the datasets with higher accuracy are larger and more imbalanced. For the datasets being highly imbalanced, with a prevalence from 0.51 to 0.94, other metrics may be more

Table 7
The overall ranking of methods according to the average accuracy obtained for every article.

Method	Accuracy	Micro accuracy	Rank
Linear SVC	0.9308	0.9609	1
Ensemble Extra Tree	0.9272	0.9604	2
Gradient Boosting	0.9262	0.9603	3
BaggingClassifier	0.9239	0.9587	4
Random Forest	0.9238	0.9596	5
Neural Net	0.9214	0.9533	6
AdaBoost	0.9154	0.9483	7
Decision Tree	0.9056	0.9445	8
K-Neighbors	0.9003	0.9003	9
Extra Tree	0.8818	0.9221	10
Bernoulli Naive Bayes	0.8591	0.8977	11
Multinomial Naive Bayes	0.8568	0.8985	12
Average	0.9060	0.9387	

Table 8
The overall ranking of methods according to the average F1-score obtained for every article.

Method	F1 score	Rank
Linear SVC	0.9266	1
Gradient Boosting	0.9214	2
BaggingClassifier	0.9202	3
Ensemble Extra Tree	0.9194	4
Random Forest	0.9148	5
Neural Net	0.9131	6
AdaBoost	0.9130	7
Decision Tree	0.9039	8
K-Neighbors	0.8792	9
Extra Tree	0.8772	10
Bernoulli Naive Bayes	0.8242	11
Multinomial Naive Bayes	0.8207	12
Average	0.8945	0.9336

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 11 best results are obtained on descriptive features combined to bag-of-words. BoW only is the best flavor for article 2, 11 and 14, whereas descriptive features alone is never the best. This seems to indicate that combining information from different sources improves the overall results.

Fig. 4 displays the normalized confusion matrix for the best methods. The normalization is done per line and it 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 0.92 to 1.00, with an average of 97.63. On the contrary, the true positive rate is lower, ranging from 0.29 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 despite the classes being highly imbalanced, the algorithms are capable of producing models that are fairly balanced.

Additionally, we provide the values of the MCC in Table 5 and F1-score in Table 6. As results are similar between both indicators, we analyze only the MCC. The MCC is generally superior to the accuracy because it takes into account the class prevalence. Therefore, it is a better metric to estimate model quality than accuracy. The MCC ranges from 0.3287 – on article 34 to 0.8890 – on article 8. The best score is not obtained by the same article as for the accuracy (article 8 achieved 95.74% accuracy, below the micro-average and just above the average). Interestingly, the MCC reveals that the performances on article 34 are rather poor in comparison to the other articles. The best methods are consistent with the accuracy counterpart. However, Linear SVC ranks first for 5 out of 11 articles, which comforts previous studies that used only Linear SVC.

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 and it is particularly true for articles with a small number of cases such as article 34.

Tables 7 and 8 rank the methods according to the average accuracy, F1-score and MCC performed on all articles. For each article and method, we kept only the best accuracy among the three dataset flavors.

In Table 9 is reported the F1-Score on article 2 for all methods. In general, all methods performs above the prevalence except for Naive Bayes methods and KNN. The score for descriptive features only is always lower than the Bag-of-Words representation, but it is not clear if descriptive features improve the results. For instance, the result on Extra Tree is higher for Bag-of-Words only and does not appear to be significant for Linear SVC. Similar results are observed on all articles (see Supplementary Material).

Table 9
F1-score for all methods on article 2.

Prev = 0.8648	F1 score - 2		
	Desc	BoW	Both
AdaBoost	0.8978	0.9452	0.9411
BaggingClassifier	0.8889	0.9438	0.9584
Bernoulli Naive Bayes	0.8022	0.8230	0.8006
Decision Tree	0.8725	0.9227	0.9464
Ensemble Extra Tree	0.8617	0.9629	0.9641
Extra Tree	0.8556	0.9295	0.9031
Gradient Boosting	0.8808	0.9565	0.9593
Linear SVC	**0.9185**	**0.9659**	**0.9661**
Multinomial Naive Bayes	0.8196	0.8202	0.7998
Neural Net	0.8933	0.9194	0.9311
Random Forest	0.8579	0.9626	0.9644

Fig. 5 displays the learning curves obtained for the best method for each article. The training error becomes (near) zero on every instance after the model has been trained with only few examples. The test error converges rather fast and remains relatively far from the training error, which is synonym of high bias. The two abovementioned observations indicate underfitting. Similar results are observed for all methods on all articles. 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. An exploratory analysis of the datasets may also help in removing some noise and finding the best predictors.

Finally, we used a Wilcoxon signed-rank test at 5% to compare the accuracy obtained on the BoW representation to the one obtained on the BoW combined with descriptive features. The difference between the samples has been found to be significant only for article 3, 6 and 8. The best result obtained on BoW 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 results are significant for every method.

In conclusion, the datasets for binary classification demonstrated a strong predictability power. Apart from article 13 and 34, each article seems to provide similar results, independently of the relatively different prevalence. The accuracy is rather high and a more informative metric, such as MCC, shows that there are still margins of improvements. Hyperparameter tuning [27] 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 as shown by Tables 4 and 5.

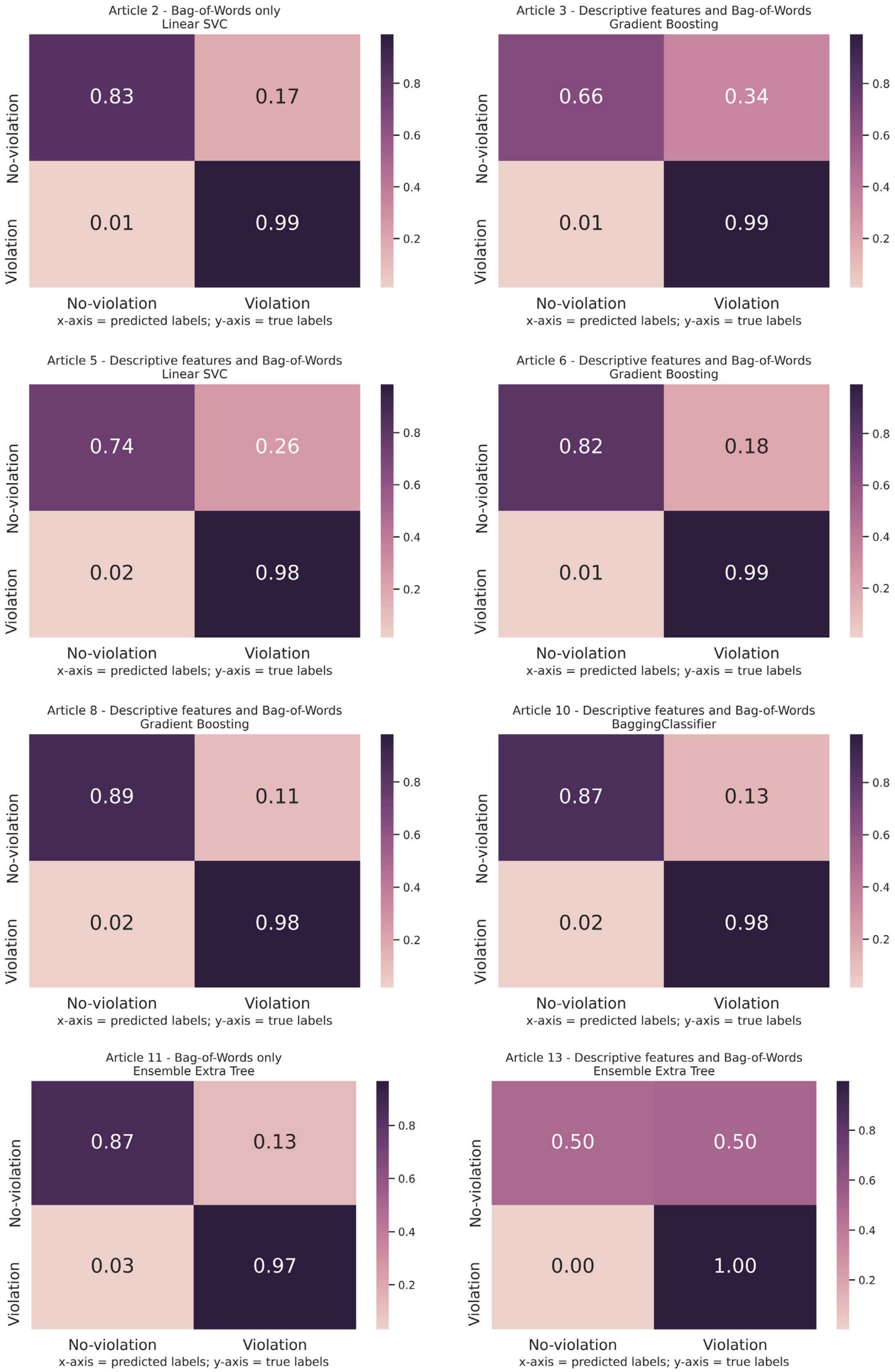


Fig. 4. Normalized Confusion Matrices for the best methods as described by Table 4.

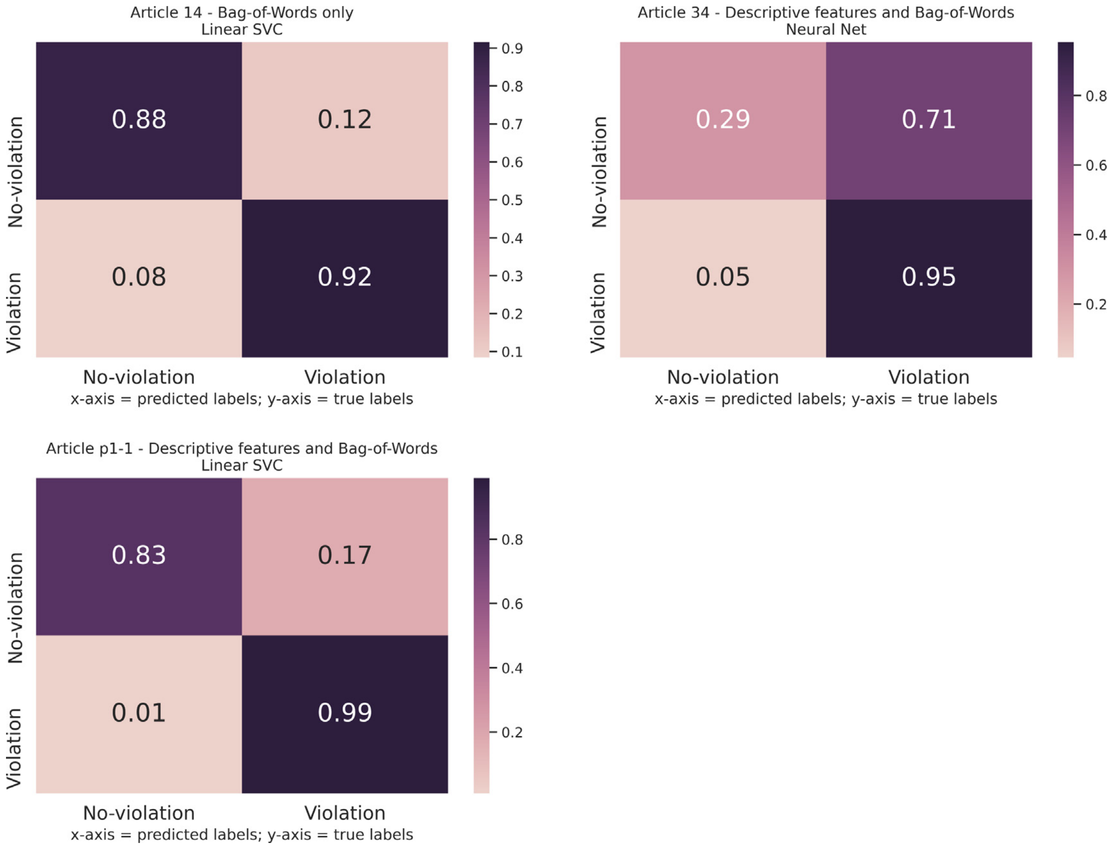


Fig. 4. (*continued*).

5.3. Discussion

To sum up, we achieved an average accuracy of 94% which is respectively 15pp and 19pp higher than best results reported in [10] and [11]. The average F1-Score obtained is 92.9 which is higher than the 82.0 obtained in [3]. However, it is worth to notice that we used traditional Machine Learning that requires less computational effort and time than state of the art Deep Learning method such as BERT. The size of the dataset does matter since we showed that the model underfits thanks to Fig. 5. 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 [28]. 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.

To sum up, the approach discussed here is more neutral in the sense we do not change the label distribution, and it still offers a robust classifier. This is confirmed in Section 7 on multilabel classification.

Our in-depth experimental evaluation 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 (obtained among all methods). We emphasize the best method 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 [27] 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. Experiments: Multiclass classification

In the previous section, we showed that most methods could obtain an accuracy higher than the dataset prevalence, and more generally, good performance metrics. Usually algorithms for binary classification adapt relatively well to multiclass problems, however, in the case of *ECHR-DB*, the labels come by pair (violation or no-violation of a given article), which may confuse the classifiers.

The experimental protocol for these experiments is identical to the one of the previous section. For computational purposes, we dropped the worst classifier from the binary datasets, namely KNN.

6.1. Results

Table 10 presents the accuracy obtained on the multiclass dataset. The best accuracy for descriptive features only and BoW

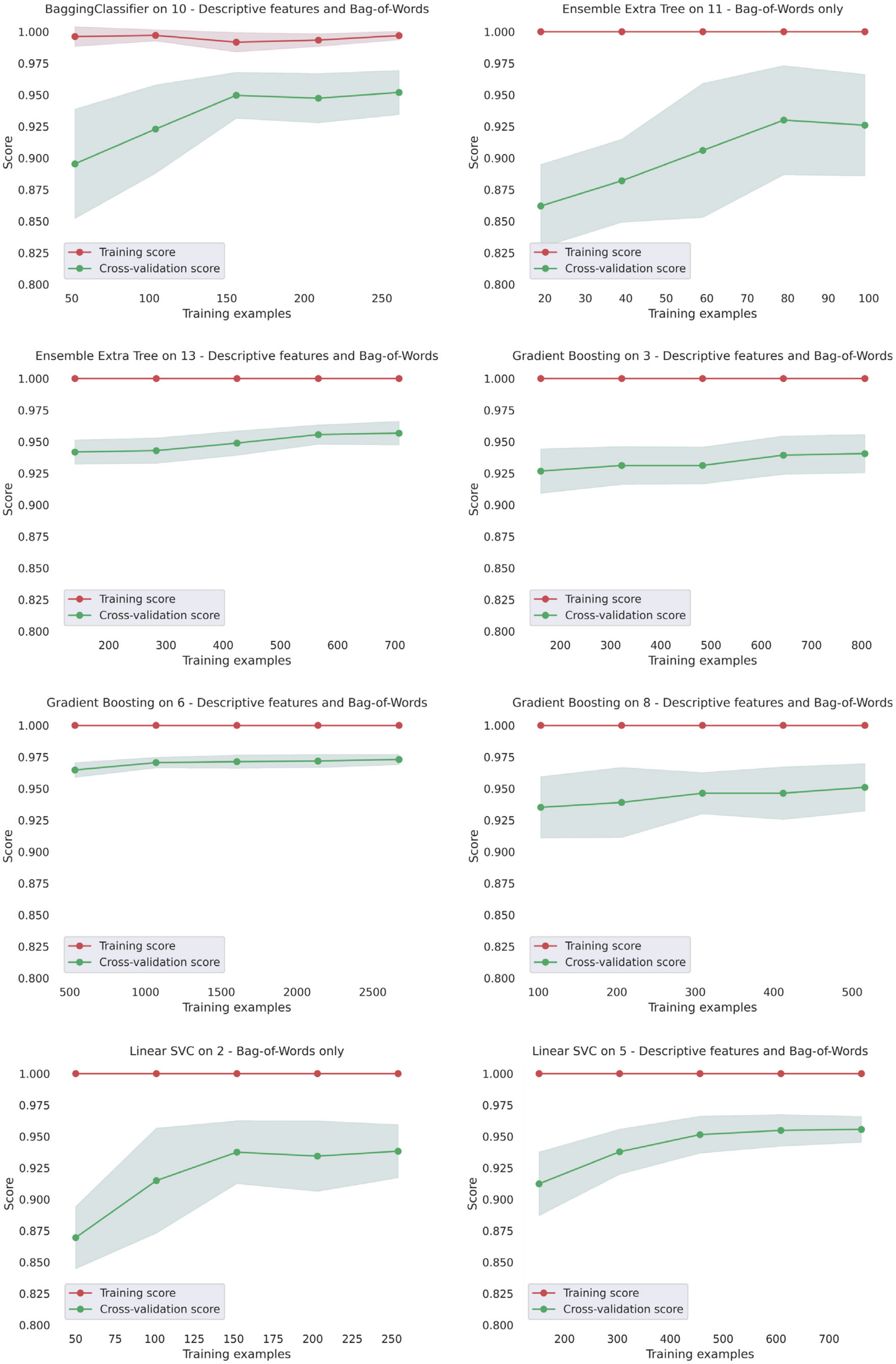


Fig. 5. Learning Curves for the best methods as described by Table 4.

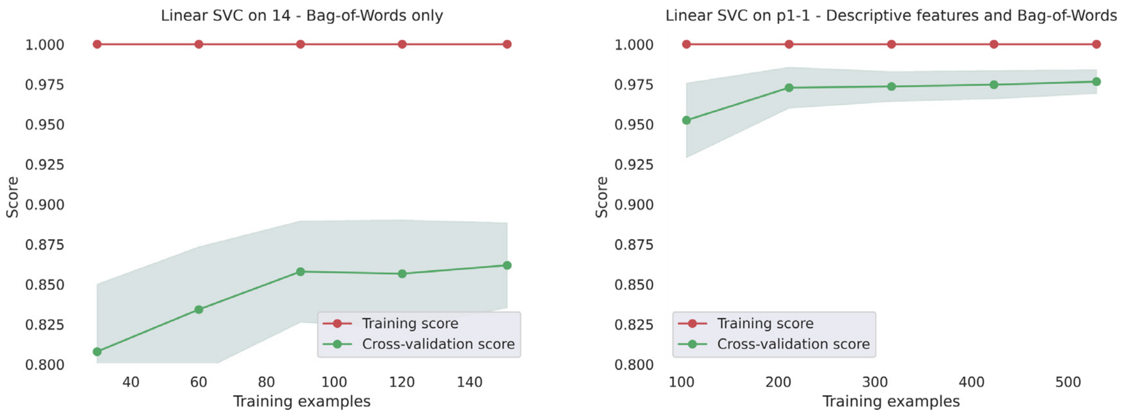


Fig. 5. (*continued*).

Table 10
The accuracy obtained for each method on the multiclass dataset.

	Accuracy — Multiclass		
	Desc	BoW	Both
AdaBoost	0.7310 (0.06)	0.6364 (0.05)	0.7000 (0.19)
BaggingClassifier	0.9140 (0.01)	0.9073 (0.01)	**0.9720** (0.01)
Bernoulli Naive Bayes	0.5460 (0.01)	0.7655 (0.01)	0.7627 (0.02)
Decision Tree	0.9078 (0.01)	0.8779 (0.01)	0.9606 (0.01)
Ensemble Extra Tree	0.8991 (0.01)	0.9139 (0.01)	0.9323 (0.01)
Extra Tree	0.8018 (0.02)	0.7548 (0.02)	0.7708 (0.02)
Linear SVC	**0.9370** (0.01)	**0.9362** (0.01)	0.9592 (0.01)
Multinomial Naive Bayes	0.8136 (0.01)	0.8022 (0.02)	0.8015 (0.01)
Neural Net	0.8971 (0.01)	0.9294 (0.01)	0.9427 (0.01)
Random Forest	0.8907 (0.01)	0.9060 (0.01)	0.9267 (0.01)

Table 11
Matthew Correlation Coefficient obtained for each method on the multiclass dataset.

	MCC — Multiclass		
	Desc	BoW	Both
AdaBoost	0.6403 (0.08)	0.5270 (0.05)	0.6380 (0.18)
BaggingClassifier	0.8819 (0.01)	0.8724 (0.02)	**0.9617** (0.01)
Bernoulli Naive Bayes	0.2653 (0.02)	0.6853 (0.02)	0.6675 (0.02)
Decision Tree	0.8734 (0.01)	0.8322 (0.01)	0.9462 (0.01)
Ensemble Extra Tree	0.8606 (0.01)	0.8815 (0.02)	0.9071 (0.02)
Extra Tree	0.7250 (0.02)	0.6631 (0.03)	0.6847 (0.03)
Linear SVC	**0.9129** (0.01)	**0.9126** (0.01)	0.9466 (0.01)
Multinomial Naive Bayes	0.7367 (0.01)	0.7363 (0.02)	0.7278 (0.02)
Neural Net	0.8578 (0.01)	0.9028 (0.01)	0.9213 (0.01)
Random Forest	0.8489 (0.01)	0.8704 (0.02)	0.8991 (0.02)

Legend: *desc* — descriptive features; *BoW* — bag of words.

only is linear SVC with 93.70% and 93.62% correctly labeled cases, respectively. This is aligned with the results obtained on binary datasets. However, the top score of 97.20% is obtained by Bagging Classifier that only ranked 3th on binary datasets. In other words, SVM ranked first on two types of features individually, but the improvement of combining the features is lower than the one obtained by Bagging Classifier. The same can be observed for Gradient Boosting that outperforms SVM. Except from Ada Boost, the standard deviation is mostly lower than 1%.

By looking at the MCC provided by Table 11, we clearly see that descriptives features provide better results than BoW for most methods. However, combining both types of features increases performances at the notable exceptions of Extra Tree, Multinomial Naive Bayes, and Ada Boost. This highly contrasts with the binary setting for which descriptive features were quantitatively far below textual features, in particular w.r.t. MCC. For binary datasets, *descriptive features only* was mostly scoring below the *BoW only*, for any article and any method (cf. GitHub[9]). On top of that, taking only the best result per flavor, the *descriptive features* score better than purely textual features. The explanation can be found by studying the confusion matrix.

Fig. 6 shows the normalized confusion matrix for Bagging Classifier. The normalization has been done per line, i.e. each line represents the distribution of cases according to their ground truth. For instance, on *descriptive features only*, for class *Article 11, no-violation*, 39% only were correctly classified and 64% assigned to a violation of article 11. The perfect classifier should thus have a diagonal of 1. The diagonal is equivalent to the recall for the corresponding class and the average the diagonal terms is the balanced accuracy [29].

Flavor *descriptive features only* has a sparser normalized confusion matrix than the counterpart with BoW. It can be explained by looking at the 2 × 2 blocks on the diagonal. These blocks are the normalized confusion matrix of the subproblem restricted to find a specific article. For instance, 99% of non-violation of article 6 has been labeled in one of the two classes related to article 6 (99% for a violation). In general, the classifiers on *descriptive features only* are good at identifying the article but generates a lot of false negatives, most likely due to the imbalance between violation and non-violation labels for a given article. Adding BoW to the case representation slightly lowers the accuracy on average but largely rebalances the 2 × 2 blocks on the diagonal. On the other hand, it seems that the textual information does not hold enough information to identify the article, which explains why classifiers perform in general lower on BoW only.

We performed two Wilcoxon signed-rank tests: first between the samples of results on BoW and BoW + descriptive features, then between descriptive features and BoW + descriptive features. The first result is clearly significant while the second is not significant, comforting us in the idea that for the multiclass domain, Bag-of-Words flavor alone is unlikely to give good results compared to descriptive features.

6.2. Discussion

The main conclusion to draw from the multiclass experiment is that the descriptive features are excellent at identifying the article while the text offers more elements to predict the outcome. It is not surprising since descriptive features are available before the judgment while the judgment itself discuss specifically the violation or not. However, the fact that descriptive features hold the best predictive power is compatible with the realism theory which considers that judges do not simply apply objective and neutral legal reasoning. Indeed, descriptive features has little to

[9] https://echr-od.github.io/ECHR-OD_project_supplementary_material/.

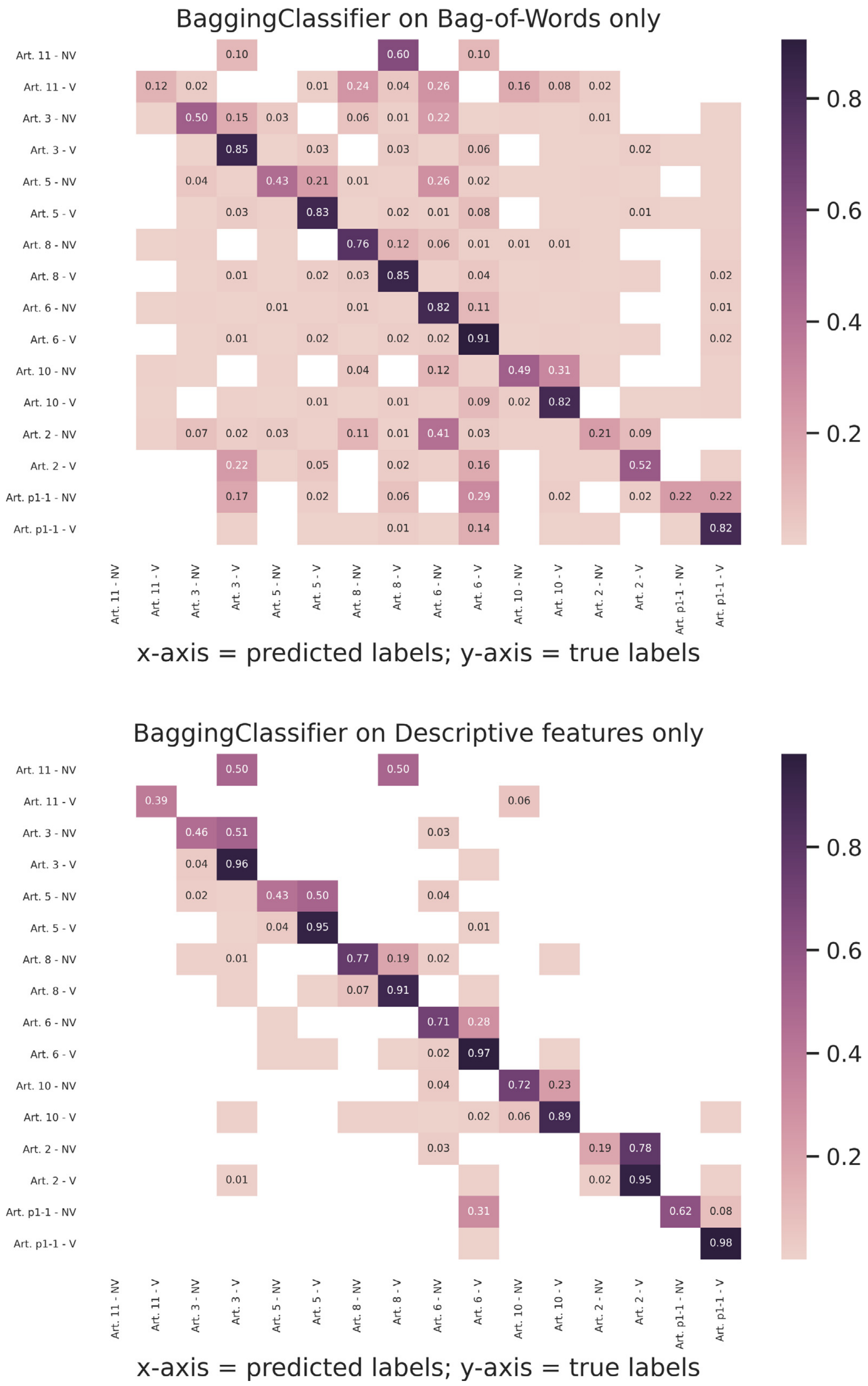


Fig. 6. Normalized Confusion Matrix for multiclass dataset. The normalization is performed per line. A white block indicates that no element has been predicted for the corresponding label. Percentages are reported only if above 1%.

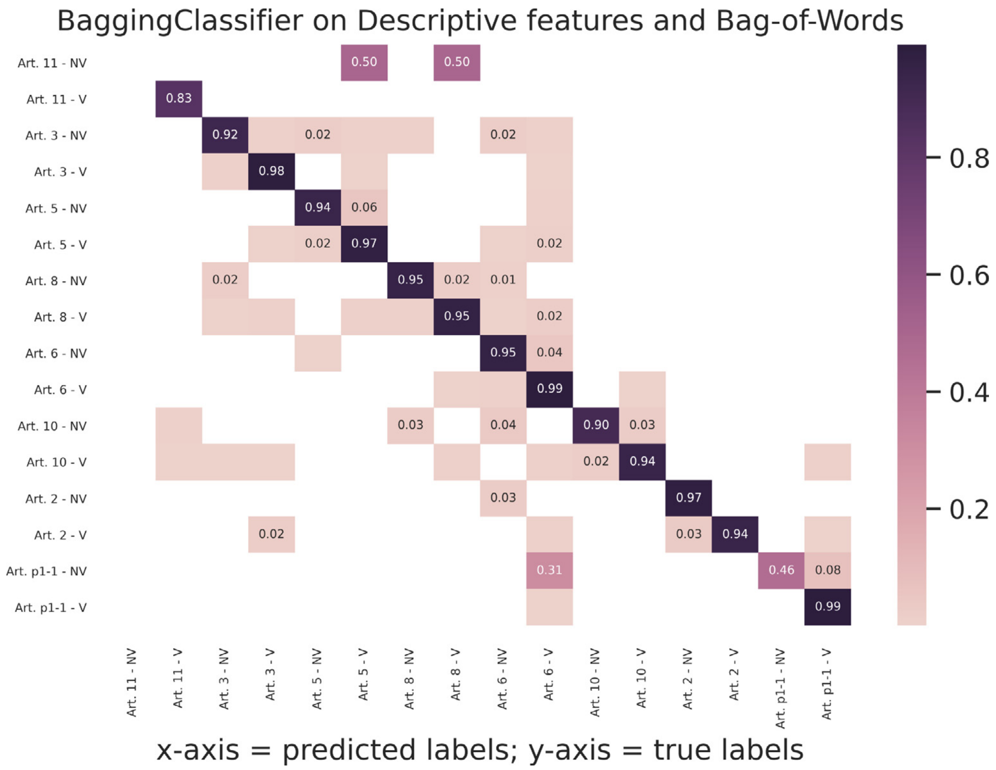


Fig. 6. (*continued*).

do with legal arguments and facts but more about judges and parties. This can be tempered by the fact that a judge might be specialized in cases related to a specific article and country, which would explain why judges and parties are strong predictors. To rule for one or the other hypothesis, we plan to explore the network of decision body members, representatives and countries provided with *ECHR-DB* in future work.

Using only BoW leads to the worst possible results, while adding textual information to the descriptive features slightly increases the accuracy and has a strong beneficial effect on discriminating between violations and non-violations. This is quite in opposition with the conclusion drawn from the experiments on the binary datasets where the textual representation clearly overperformed while the descriptive features had only a marginal effect.

This indicates that it might be more interesting to create a two-stage classifier, namely: a multiclass classifier – for determining an article based on descriptive features, followed by an article-specific classifier – for determining if the article is violated or not. Over-sampling techniques to deal with imbalanced classes constitute another axis of improvement to explore in future work.

Finally, we conclude that the benefit of combining sources of information is not monotonic: the best scoring method on individual types of features might not be the best method overall.

7. Experiments: Multilabel classification

A multilabel dataset generalizes the multiclass one in a way there is not only one article to identify before predicting the outcome, but an unknown number. In the ECHR, application must specify the articles that are to be discussed. These articles are taking into consideration to judge if a case is admissible or not. The multilabel dataset reflects a real-life situations in which a lawyer might advise the plaintiff on the articles to be added in her application, as well as the probability of violation or not. In such a case, on top of analyzing the usual performance metrics, we would like to quantify how good are the methods to identify all the articles in each case. From the multiclass results, it is expected that the textual information alone will provide the lowest results among all flavors which would be an interesting results since the judgment is obviously not known at the moment of filling the application.

7.1. Protocol

Appreciating the results of a multilabel classifier is not as easy as in the binary or multiclass case. For instance, having wrongly added one label to 100 cases is not exactly the same as adding 100 wrong labels to a single case. Similarly, being able to correctly predict at least one correct label per case, is not the same as predicting all good labels for a fraction of the cases, even if the total amount of correct labels is the same in both scenarios. The distributions of ground truth and predicted labels among the dataset are important for evaluating the quality of a model.

For this reason, we reported the following multilabel-specific metrics: subset accuracy, precision, recall, F_1-score, Hamming loss, and the Jaccard index. The subset accuracy is the strictest metric since it measures the percentage of samples such that all the labels are correctly predicted. It does not account for partly correctly labeled vectors. The Hamming loss calculates the percentage of wrong labels in the total number of labels. The Jaccard index measures the number of correctly predicted labels divided by the union of prediction and true labels.

We are interested in quantifying how much a specific article was properly identified, as well as how many cases with a given number of labels are correctly labeled, taking into account their respective prevalence in the dataset reported in Fig. 3. Indeed, about 68% of cases in the multilabel dataset have only one label such that a classifier assigning only one label to each case could reach about 68% of subset accuracy.

Not all binary classification algorithms can be extended for the multilabel problem. Therefore, in our experiments we used

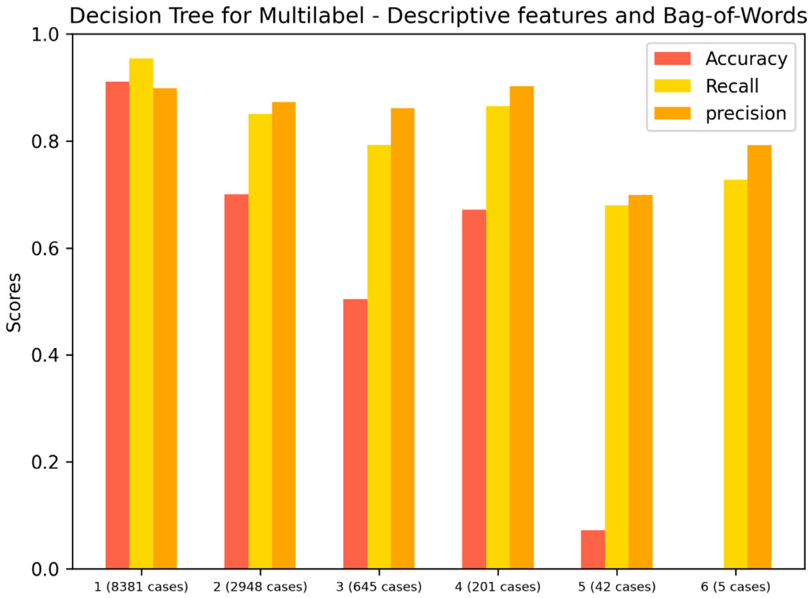


Fig. 7. Multilabel scores depending on the number of labels assigned.

Table 12
The accuracy, precision and recall for each method on the multilabel dataset.

	Accuracy			Precision			Recall		
	desc	BoW	both	desc	BoW	both	desc	BoW	both
Decision Tree	**0.8101** (0.02)	0.6720 (0.01)	**0.8337** (0.02)	0.8634	0.7900	0.8948	**0.8596**	**0.7730**	**0.8848**
Ensemble Extra Tree	0.7399 (0.02)	0.6681 (0.02)	0.6942 (0.01)	**0.8831**	**0.8836**	**0.9031**	0.7559	0.7008	0.7174
Extra Tree	0.6087 (0.02)	0.5438 (0.02)	0.5572 (0.02)	0.7311	0.6758	0.6882	0.7076	0.6554	0.6679
Neural Net	0.7357 (0.02)	**0.6844** (0.03)	0.6905 (0.02)	0.8780	0.8678	0.8703	0.7900	0.7641	0.7691
Random Forest	0.7224 (0.02)	0.6470 (0.02)	0.6688 (0.02)	0.8710	0.8740	0.8959	0.7381	0.6784	0.6896

the following five algorithms: Extra Tree, Decision Tree, Random Forest, Ensemble Extra Tree, and Neural Network. As previously, a 10-fold cross-validation has been performed on each flavor.

7.2. Results

The accuracy is reported in Table 12 and it shows that Decision Tree outperforms with 81.01% of cases that have been totally correctly labeled. Similarly to the multiclass setting, descriptive features provide a better result than BoW. Decision Tree scores also the best for the F_1-score and recall (see Table 13). However, Ensemble Extra Tree overperforms Decision Tree w.r.t. precision and F_1-score. Decision Tree provides the best results for the *strict* metrics (highest accuracy and lowest Hamming loss) but also on more permissive metrics (Jaccard index). Therefore, Decision Tree is clearly the top classifier for multilabel which is a bit surprising, since it ranked 8th over the binary datasets and 3rd on the multiclass one (see Table 7).

As expected, the BoW flavor provides the worst possible results. Similarly to the experiments on the multiclass dataset, the textual information is inefficient for identifying the article.

Fig. 7 shows the accuracy, recall, and precision depending on the number of labels assigned by Decision Tree on the test dataset. It also indicates the number of cases for each label count. It is striking to observe how the distribution of cases depending on the labels is close to the real distribution shown in Fig. 3. Therefore, we can reasonably assume that the model correctly identifies the articles of a given case. The subset accuracy for cases with a single label is consistent with the score on the multiclass dataset.

The subset accuracy decreases linearly with the number of labels, which is not surprising, since the metric becomes stricter with the number of labels. However, the recall and precision remain stable, above 80% on average, indicating that not only the algorithm carefully identifies the labels (recall) but also identify a large portion of labels (precision). Thus, from these observations, we can clearly discard the possibility that the algorithm mostly focuses on cases with a single label.

7.3. Discussion

The multilabel experiment is, as far as we know, the first experiment in the legal domain to predict a more structured outcome than a binary outcome. On top of that, it showed that, contrarily to all past studies, textual features are not necessarily holding the most adequate information for prediction. In particular, the multilabel experiment showed that *descriptive features only* are enough to obtain results that are quantitatively close to the combination of both sources of information. We are confident that this conclusion is valid despite being in opposition with all previous conclusions. Indeed, not only the corpus of documents we used is larger but also our models provide better results.

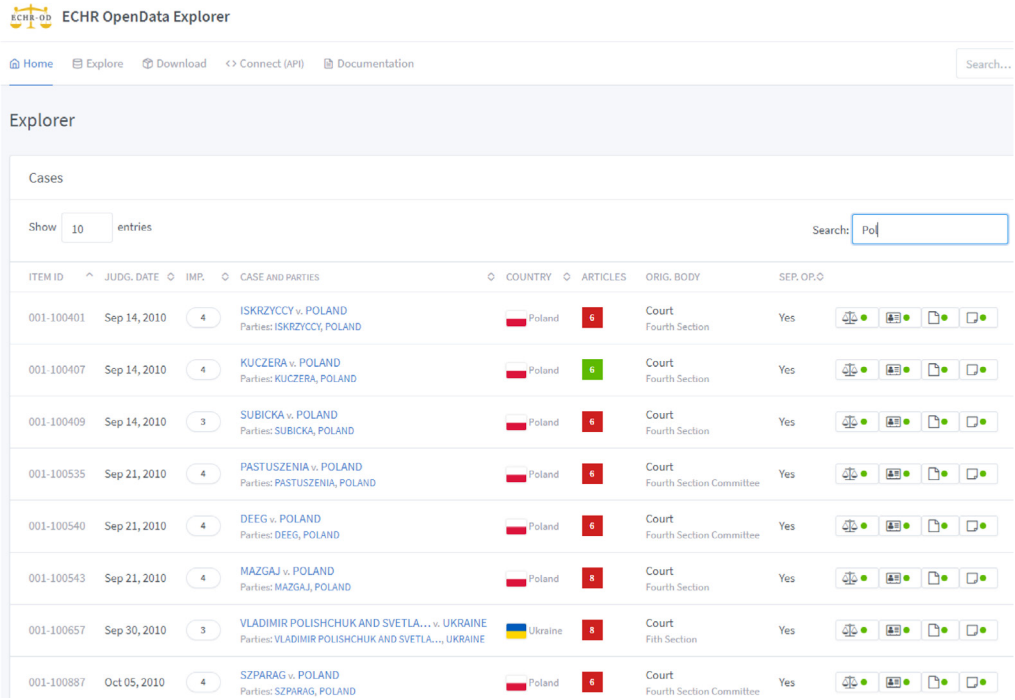
This opens the road to practical applications, while previous studies used only data known a posteriori. For instance, knowing a basic description of a case, a citizen might quickly determine the part of the law that applies to her case, and a reasonable estimation of the outcome. This might help her to find an advisor or representative specialized in this area of the law.

8. 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

Table 13
The F1-score, Jaccard index and hamming loss for each method on the multilabel dataset.

	F1 score			Jaccard index			Hamming loss		
	desc	BoW	both	desc	BoW	both	desc	BoW	both
Decision Tree	0.8634	0.7900	0.8948	**0.7807**	0.6680	**0.8202**	**0.0067**	0.0107	**0.0055**
Ensemble Extra Tree	**0.8831**	**0.8836**	**0.9031**	0.7104	0.6653	0.6900	0.0076	0.0088	0.0081
Extra Tree	0.7311	0.6758	0.6882	0.5932	0.5280	0.5406	0.0134	0.0163	0.0158
Neural Net	0.8780	0.8678	0.8703	0.7351	**0.7054**	0.7088	0.0072	**0.0080**	0.0079
Random Forest	0.8710	0.8740	0.8959	0.6941	0.6447	0.6652	0.0080	0.0094	0.0087



ECHR OpenData Explorer

Explorer

Cases

Show 10 entries

Search: Pol

ITEM ID	JUDG. DATE	IMP.	CASE AND PARTIES	COUNTRY	ARTICLES	ORIG. BODY	SEP. OP.	
001-100401	Sep 14, 2010	4	ISKRZYCCY v. POLAND Parties: ISKRZYCCY, POLAND	Poland	6	Court Fourth Section	Yes	● ● ● ●
001-100407	Sep 14, 2010	4	KUCZERA v. POLAND Parties: KUCZERA, POLAND	Poland	6	Court Fourth Section	Yes	● ● ● ●
001-100409	Sep 14, 2010	3	SUBICKA v. POLAND Parties: SUBICKA, POLAND	Poland	6	Court Fourth Section	Yes	● ● ● ●
001-100535	Sep 21, 2010	4	PASTUSZENIA v. POLAND Parties: PASTUSZENIA, POLAND	Poland	6	Court Fourth Section Committee	Yes	● ● ● ●
001-100540	Sep 21, 2010	4	DEEG v. POLAND Parties: DEEG, POLAND	Poland	6	Court Fourth Section Committee	Yes	● ● ● ●
001-100543	Sep 21, 2010	4	MAZGAJ v. POLAND Parties: MAZGAJ, POLAND	Poland	8	Court Fourth Section	Yes	● ● ● ●
001-100657	Sep 30, 2010	3	VLADIMIR POLISHCHUK AND SVETLA... v. UKRAINE Parties: VLADIMIR POLISHCHUK AND SVETLA..., UKRAINE	Ukraine	8	Court Fith Section	Yes	● ● ● ●
001-100887	Oct 05, 2010	4	SZPARAG v. POLAND Parties: SZPARAG, POLAND	Poland	6	Court Fourth Section Committee	Yes	● ● ● ●

Fig. A.8. Listing of cases with real time sorting and search. Each row provides the judgment date, the parties, the country(ies), the main conclusion, and the information whether the raw judgment file and processed documents are available.

provide cleaned and transformed content from the repository of the European Court of Human Rights, that is ready to be used 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. The repository will be iteratively corrected and updated along with the European Court of Human Rights new judgments.

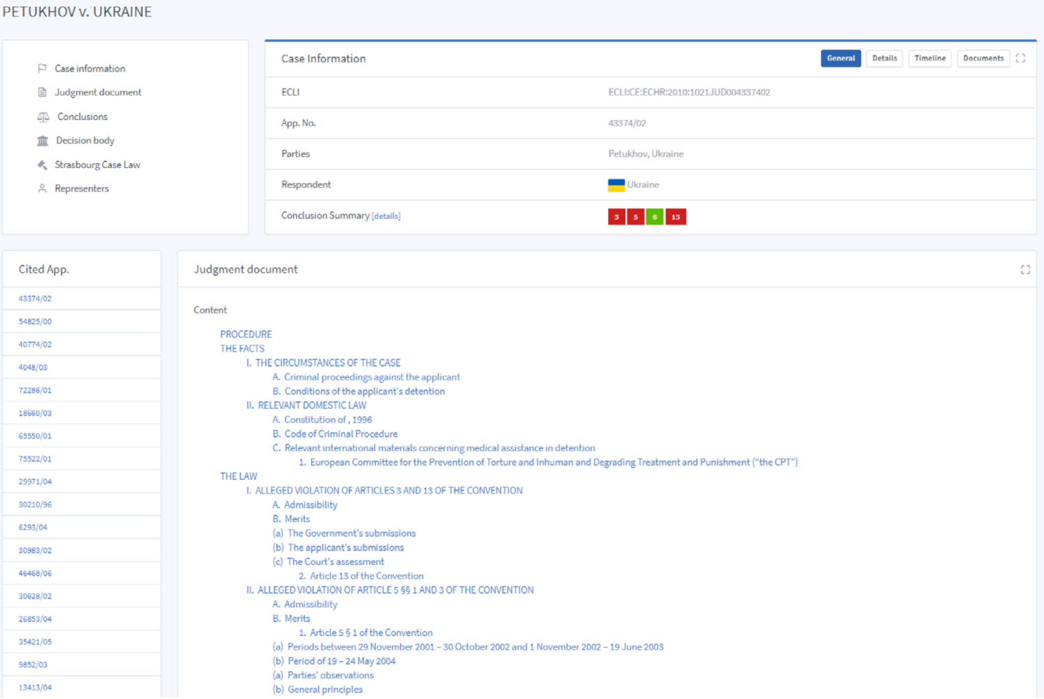
Currently, *ECHR-DB* is the largest and most exhaustive repository of legal documents from the European Court of Human Rights. It includes several types of data that can be easily used to reproduce various experiments, which have been done so far by other researchers. We argue that providing the final data is not enough to ensure quality and trust. In addition, there are always some alternative 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 pipeline of dataset construction from scratch. The pipeline was implemented by means of Python scripts and available on GitHub.[10]

The experiments on *ECHR-DB* provide a 15pp improvement compared to the previous studies on binary classification and similar results than the state of the art Deep Learning. They also allow us to draw the following conclusions.

1. The models for binary classification clearly underfit while the data is already exhaustive. Therefore, to provide a larger training set, we need to consider the more complex multiclass or multilabel problem. Despite this additional complexity, the multiclass and multilabel models provide as good results as the binary counterparts thanks to this larger training set.
2. Textual features are good at finding if there is a violation or not for a given article, while the descriptive features alone are good at identifying the article. Descriptive features surprisingly hold reasonable predictive power.
3. For the most complex problem that is the multilabel setting, using *descriptive features only* provides equivalent results as textual features. This is particularly important, since descriptive features are available mostly before a verdict contrarily to the judgment document, by definition available after. This opens the road to more practical applications, especially if the results are reproducible with any type of judgments, beyond the European Court of Human Rights.

The experiments indicated several axes of improvements, e.g., better embedding with state of the art encoders, hyperparameter tuning, multi-stage classifiers, and transfer learning. From the obtained 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, other interesting research questions and problems arise from the proposed repository, e.g. *can*

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



Case Information

ECLI	ECLI:CE:ECHR:2010:1021JUD004337402
App. No.	43374/02
Parties	Petukhov, Ukraine
Respondent	Ukraine
Conclusion Summary [details]	3 5 6 13

Cited App.

43374/02
54825/00
40774/02
4048/03
72286/01
18660/03
65550/01
75522/01
29971/04
30210/96
6293/04
30983/02
46468/06
30628/02
26853/04
35421/05
9852/03
13413/04

Judgment document

Content

```
PROCEDURE
THE FACTS
    I. THE CIRCUMSTANCES OF THE CASE
        A. Criminal proceedings against the applicant
        B. Conditions of the applicant's detention
    II. RELEVANT DOMESTIC LAW
        A. Constitution of , 1996
        B. Code of Criminal Procedure
        C. Relevant international materials concerning medical assistance in detention
            1. European Committee for the Prevention of Torture and Inhuman and Degrading Treatment and Punishment ("the CPT")
THE LAW
    I. ALLEGED VIOLATION OF ARTICLES 3 AND 13 OF THE CONVENTION
        A. Admissibility
        B. Merits
        (a) The Government's submissions
        (b) The applicant's submissions
        (c) The Court's assessment
            2. Article 13 of the Convention
    II. ALLEGED VIOLATION OF ARTICLE 5 §§ 1 AND 3 OF THE CONVENTION
        A. Admissibility
        B. Merits
            1. Article 5 § 1 of the Convention
        (a) Periods between 29 November 2001 – 30 October 2002 and 1 November 2002 – 19 June 2003
        (b) Period of 19 – 24 May 2004
        (a) Parties' observations
        (b) General principles
```

Fig. A.9. Partial display of a case. In particular, the cited applications are extracted. The table of content shows the tree nature of a judgment document.

we provide legal justification in natural language to a prediction?, which will be addressed in the future work.

Last but not least, we encourage all researchers to explore the data, generate new datasets for various problems and submit their contributions to the project.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Additional features of *ECHR-DB*

In addition to the final database and files of *ECHR-DB*, a portal has been developed with two main features: (1) an online explorer to browse cases and (2) an API to interface the database with external applications.

A.1. The explorer

The explorer is a web application that allows to sort and search cases in real time and display every information gathered about a specific case, including the members of the decision body, the timeline, associated documents and detailed conclusion. Additionally, the judgment document is displayed as a tree and the cross-citations are extracted from each document. Figs. A.8 and A.9 show the interface displaying the list of cases and a particular case, respectively.

Note that the explorer always uses the SQL database available, such that the explorer is always up to date with the latest version of the database.

A.2. REST API

The standardized REST API provides a convenient programmatic way to retrieve data from *ECHR-DB*. The API allows to download specific documents, access cases, parties, representatives, citations, and conclusions. The documentation is available at https://echr-opendata.eu/api/v1/docs and it allows to try any request. Fig. A.10 shows the documentation interface and most of the available endpoints.

Two examples of manual API calls are provided below. The first one returns the version of *ECHR-DB* used by the API, and the second returns the list of documents available for a specific case and how to download them.

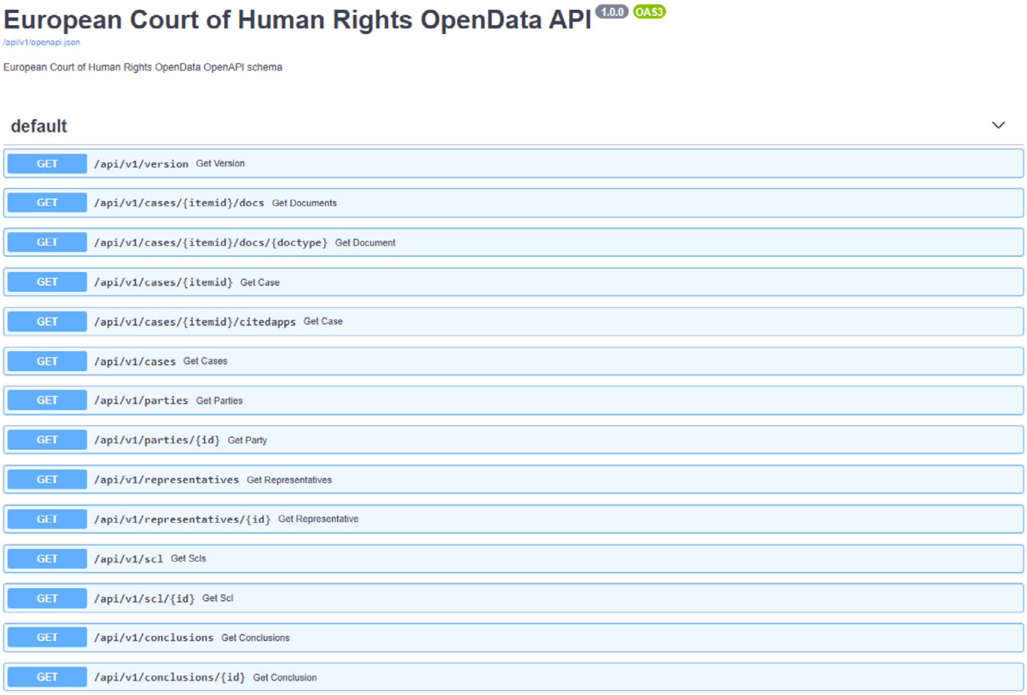
```
curl -X GET "https://echr-opendata.eu/api/v1/version" -H
        "accept: application/json"

"2.0.0"

curl -X GET "https://echr-opendata.eu/api/v1/cases/001-100018/docs"
         -H "accept: application/json"

{
  "judgment": {
    "available": true,
    "uri": "/api/v1/cases/001-100018/docs/judgment"
  },
  "bow": {
    "available": true,
    "uri": "/api/v1/cases/001-100018/docs/bow"
  },
  "tfidf": {
    "available": true,
    "uri": "/api/v1/cases/001-100018/docs/tfidf"
  },
  "parsed_judgment": {
    "available": true,
    "uri": "/api/v1/cases/001-100018/docs/parsed_judgment"
  }
}
```

Therefore, to download the Bag-of-Words representation of case 001-100018, it is enough to call:



European Court of Human Rights OpenData API 1.0.0 OAS3

/api/v1/openapi.json

European Court of Human Rights OpenData OpenAPI schema

default

```
GET /api/v1/version Get Version
GET /api/v1/cases/{itemid}/docs Get Documents
GET /api/v1/cases/{itemid}/docs/{doctype} Get Document
GET /api/v1/cases/{itemid} Get Case
GET /api/v1/cases/{itemid}/citedapps Get Case
GET /api/v1/cases Get Cases
GET /api/v1/parties Get Parties
GET /api/v1/parties/{id} Get Party
GET /api/v1/representatives Get Representatives
GET /api/v1/representatives/{id} Get Representative
GET /api/v1/scl Get Scls
GET /api/v1/scl/{id} Get Scl
GET /api/v1/conclusions Get Conclusions
GET /api/v1/conclusions/{id} Get Conclusion
```

Fig. A.10. Documentation of the REST API. Each endpoint can be tested online.

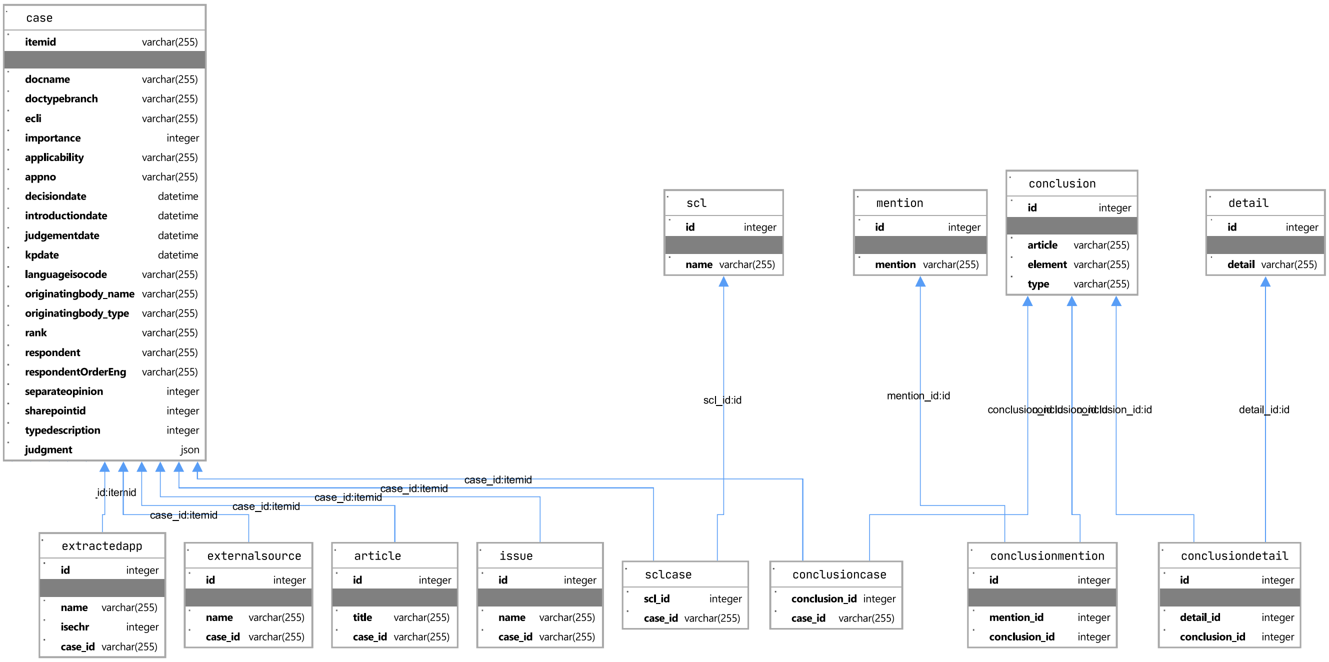


Fig. B.11. Relational schema of the SQL database (part I)

```
curl -X GET "https://echr-opendata.eu/api/v1/cases/001-100018/docs/bow"
```

Appendix B. Relational schema

This Appendix contains the relation schema of the SQL database. A full size version is available at https://osf.io/2tszn/ (see Figs. B.11 and B.12).

Appendix C. Supplementary material

Supplementary Material contains the detailed results of all experiments. In particular, it includes the results for each model on each article for all metrics.

The Supplementary Material is available at: https://echr-od.github.io/ECHR-OD_project_supplementary_material

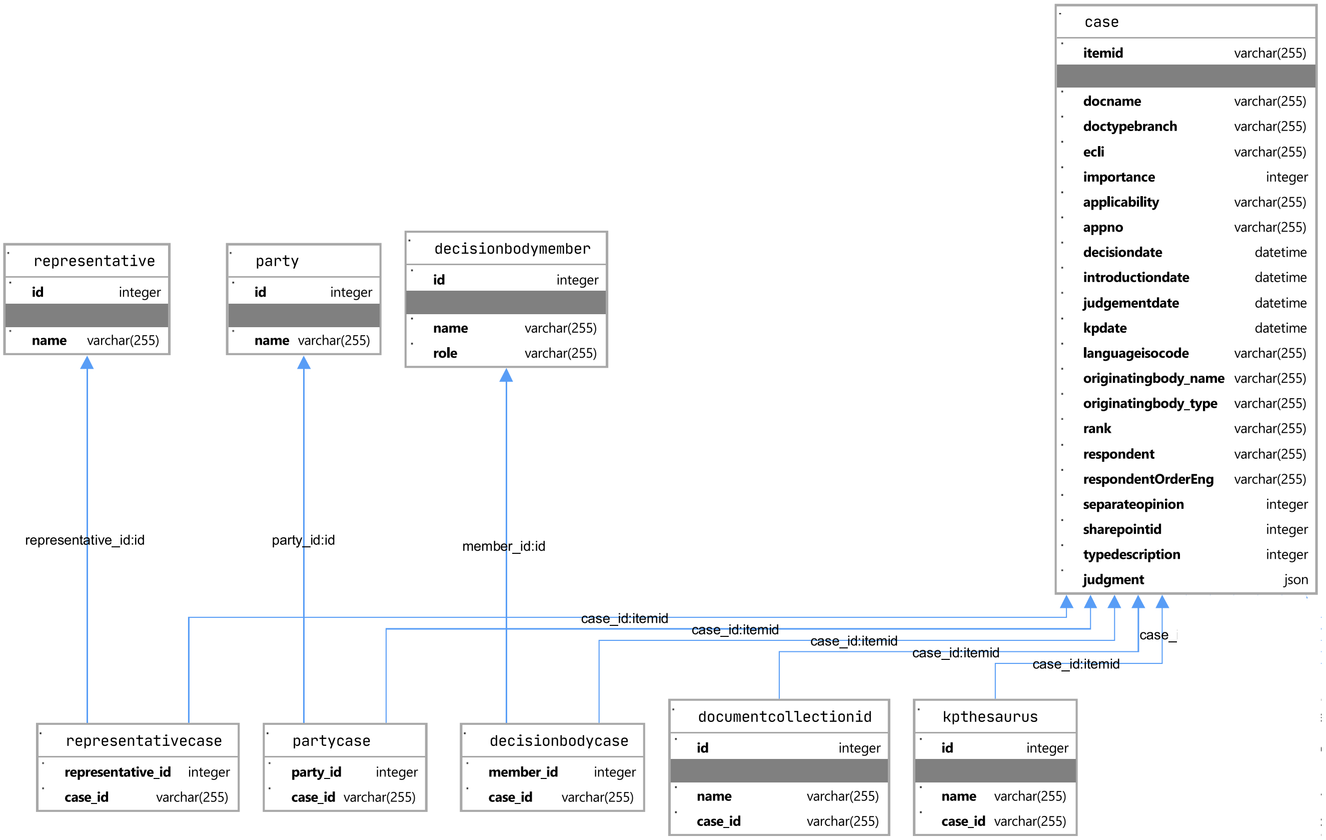


Fig. B.12. Relational schema of the SQL database (part II)

References

[1] Jack G. Conrad, L.K. Branting, Introduction to the special issue on legal text analytics, Artif. Intell. Law 26 (2) (2018) 99–102.

[2] G. Antoniou, G. Baryannis, S. Batsakis, G. Governatori, L. Robaldo, G. Siragusa, I. Tachmazidis, Legal reasoning and big data: opportunities and challenges, in: Proc. of Workshop on MIning and REasoning with Legal texts (MIREL), 2018.

[3] I. Chalkidis, I. Androutsopoulos, N. Aletras, Neural legal judgment prediction in english, 2019, pp. 4317–4323.

[4] R. Cichowski, Chrun, European court of human rights database, version 1.0 release 2017, 2017, http://depts.washington.edu/echrdb/.

[5] A. Quemy, R. Wrembel, On integrating and classifying legal text documents, in: International Conference on Database and Expert Systems Applications (DEXA), Vol. 12391, 2020.

[6] T.W. Ruger, P.T. Kim, A.D. Martin, K.M. Quinn, The supreme court forecasting project: Legal and political science approaches to predicting supreme court decisionmaking, Columbia Law Rev. 104 (4) (2004) 1150–1210.

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

[8] A.D. Martin, K.M. Quinn, T.W. Ruger, P.T. Kim, Competing approaches to predicting supreme court decision making, Perspect. Politics 2 (4) (2004) 761–767.

[9] R. Guimerà, M. Sales-Pardo, Justice blocks and predictability of U.S. Supreme Court votes, PLoS ONE 6 (11) (2011) e27188.

[10] N. Aletras, D. Tsarapatsanis, D. Preoţiuc-Pietro, V. Lampos, Predicting judicial decisions of the european court of human rights: a natural language processing perspective, PeerJ Comput. Sci. 2 (2016) e93.

[11] M. Medvedeva, M. Vols, M. Wieling, Using machine learning to predict decisions of the european court of human rights, Artif. Intell. Law (2019) URL https://doi.org/10.1007/s10506-019-09255-y.

[12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.

[13] K. Atkinson, T. Bench-Capon, Reasoning with legal cases: Analogy or rule application? in: Proc. of Int. Conf. on Artificial Intelligence and Law (ICAIL), ACM, 2019, pp. 12–21.

[14] T. Bench-Capon, The need for good old fashioned ai and law, International Trends in Legal Informatics: A Festschrift for Erich Schweighofer.

[15] A. Quemy, Data science techniques for law and justice: Current state of research and open problems, in: Proc. of Workshops European Conf. on Advances in Databases and Information Systems, in: CCIS, vol. 767, Springer, 2017, pp. 302–312.

[16] V. Aleven, K.D. Ashley, Evaluating a learning environment for case-based argumentation skills, in: International Conference on Artificial Intelligence and Law (ICAIL), ACM, New York, NY, USA, 1997, pp. 170–179.

[17] P.M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence 77 (1995) 321–357.

[18] P.M. Dung, P.M. Thang, Towards (probabilistic) argumentation for jury-based dispute resolution, in: Conference on Computational Models of Argument (COMMA), 2010, pp. 171–182.

[19] P.M. Dung, P.M. Thang, Towards an argument-based model of legal doctrines in common law of contracts, 7 (2008) 111–126.

[20] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.

[21] A. Quemy, Predictions of the european court of human rights, 2019, https://github.com/aquemy/ECHR-OD_predictions.

[22] S.M.F. Ali, R. Wrembel, From conceptual design to performance optimization of ETL workflows: current state of research and open problems, VLDB J. 26 (6) (2017) 777–801.

[23] E. Loper, S. Bird, Nltk: The natural language toolkit, in: Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, Association for Computational Linguistics, Philadelphia, 2002.

[24] R. Řehůřek, P. Sojka, Software framework for topic modelling with large corpora, in: Proc. of Worksh. on New Challenges for NLP Frameworks, ELRA, 2010, pp. 45–50.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[26] D. Chicco, Ten quick tips for machine learning in computational biology, BioData Min. 10 (1) (2017).

[27] A. Quemy, Two-stage optimization for machine learning workflow, Inf. Syst. 92 (2020) 101483, http://dx.doi.org/10.1016/j.is.2019.101483.

[28] P. Lemberger, I. Panico, A primer on domain adaptation, 2020, arXiv:2001.09994.

[29] K.H. Brodersen, C.S. Ong, K.E. Stephan, J.M. Buhmann, The balanced accuracy and its posterior distribution, in: Proc. of Int. Conf. Pattern Recognition, IEEE, 2010, pp. 3121–3124.