Article 700 Identification in Judicial Judgments: Comparing Transformers and Machine Learning Models

Sid Ali Mahmoudi, Charles Condevaux, Guillaume Zambrano, Stéphane Mussard

stats

MDPI

Communication

Article 700 Identification in Judicial Judgments: Comparing Transformers and Machine Learning Models

Sid Ali Mahmoudi, Charles Condevaux, Guillaume Zambrano and Stéphane Mussard *

Université Nîmes Chrome, Avenue du Dr. Georges Salan, 30000 Némes, France;
sid.mahmoudi@unimes.fr (S.A.M.); charles.condevaux@unimes.fr (C.C.); guillaume.zambrano@unimes.fr (G.Z.)
* Correspondence: stephane.mussard@unimes.fr

Abstract: Predictive justice, which involves forecasting trial outcomes, presents significant challenges due to the complex structure of legal judgments. To address this, it is essential to first identify all claims across different categories before attempting to predict any result. This paper focuses on a classification task based on the detection of Article 700 in judgments, which is a rule indicating whether the plaintiff or defendant is entitled to reimbursement of their legal costs. Our experiments show that conventional machine learning models trained on word and document frequencies can be competitive. However, using transformer models specialized in legal language, such as *Judicial CamemBERT*, also achieves high accuracies.

Keywords: CamemBERT; text classification; predictive justice; TF-IDF

Citation: Mahmoudi, S.A.; Condevaux, C.; Zambrano, G.; Mussard, S. Article 700 Identification in Judicial Judgments: Comparing Transformers and Machine Learning Models. *Stats* **2024**, *7*, 1421–1436. https://doi.org/10.3390/stats7040083

Academic Editor: Wei Zhu

Received: 14 August 2024
Revised: 2 October 2024
Accepted: 7 November 2024
Published: 26 November 2024

1. Introduction

Predictive justice currently constitutes a main discipline lying at the intersection between legal science and computer science. In recent years, the related literature has evolved significantly because of several needs. First of all, the increasing number of cases means that the justice system needs to be automated, for very simple cases, to relieve the pressure on the courts, even if judges may be reluctant to accept this kind of practice. Secondly, from the lawyers' point of view, the increase in litigation and the new types of litigation (linked for example to the protection of personal data) encourage the automated processing of certain tasks: finding judgments in cases similar to that of a new customer who sets out the facts of their dispute, and predicting the outcome of the dispute to better inform their customer, predicting the amounts of damages that the judge is likely to award. These growing needs, emanating from the litigant, must be accompanied by legal experts. However, in certain countries such as France, the number of lawyers remains very low. The Paris Bar indicates that France has only 102.6 lawyers per 100,000 inhabitants on average, while other countries such as Germany have three times as many lawyers [1]. Therefore, the use of automated predictive justice processes is becoming a major concern.

The prediction of the judge's outcome can be made either ex-ante or ex-post. Ex-post models are based on a corpus of previous judgments, while machine learning (ML) models are trained on the "facts" section of these judgments, see for instance [2]. This involves using vector representations of words and phrases to predict the outcome. The ex-ante approach consists of predicting the outcome before the judgment is made, by analyzing the arguments of the lawyers and any other accessible information obtained before the trial, for example [3] who use facts available before litigation of the European Court of Justice. In each case, ML models need an important quantity of information. However, the corpus may be reduced to particular cases related to precise claims to avoid too much heterogeneity in the data.

A *case* is defined to be a situation in which two persons (at least) are involved in litigation, based on a *claim*, such as divorce, personal injury, or moral damages. To predict

a case's judicial outcome, it is first necessary to indicate the claim category to which the case belongs. Indeed, judicial litigation exists if a claimant (*the plaintiff*) mobilizes a court through his or her lawyer to prosecute the person he or she believes to be guilty (*the defendant*).

A basic hypothesis in the jurimetric literature is to predict the outcome of litigation with deep learning or ML models trained on judicial documents (*judgments*) related to a given claim category. This leads the models to point out some particular patterns such as sentences and precise vocabulary associated with a category of claim [4]. This hypothesis has been shown to be verified in practice, see for example [5].

In this respect, it is therefore necessary to build upstream models that detect the categories of claims, based on judgments annotated by legal experts. This is precisely the aim of this paper. These models must recognize a discriminant set of words. For this purpose, the well-known TF-IDF technique based on word or expression frequencies, proposed by [6], can be used. In addition, transformer models [7] can work very well since these models use the context in sentences to solve various tasks: classification, question answering, named entity recognition, sentiment analysis, etc. Using transfer learning, we take benefit from an already existing transformer to specialize it on a classification task, that of recognizing the `Article 700` in a judgment. The `Article 700` holds significant importance in the legal French proceedings, as it permits the prevailing party in a dispute to recover part or all of the legal costs incurred. Therefore, locating the presence of `Article 700` can be an excellent strategy for discovering the outcome of a dispute.

The aims of our paper are the following:

1. To show that baseline models based on word frequencies can find `Article 700` and sentences related to this claim with high accuracy. While simplistic, the (baseline) binary classification models utilizing TF-IDF vectorization can outperform the results achieved by transformer-based models.
2. To show that a transformer based on bidirectional encoders, fine-tuned on judicial vocabulary for a classification task, can reach excellent accuracy around 0.99 indicating that the identification task is solved.

The paper is organized as follows. Section 2 analyses the importance of the task underlying the `Article 700`. Section 3 outlines the methodology to detect the presence of the `Article 700` in judgments, that is, either the use of models based on TF-IDF or transformers. Section 4 analyzes the results of the classification task by comparing the baseline to the transformers. Section 5 closes the paper.

2. Motivations

`Article 700` claims hold significant relevance in the French legal system [8], as they directly impact judicial decisions regarding the allocation of legal costs and fees. These claims allow judges to decide whether procedural costs will be partially or fully reimbursed. A favorable ruling under `Article 700` often implies a victory in the primary claim, such as damages or dismissal, making the outcome of this article a predictive tool for legal professionals. Lawyers can leverage this predictive capacity to better advise their clients, especially regarding potential financial gains from legal cost recovery. Moreover, the ability to anticipate decisions on `Article 700` claims can enhance a lawyer's strategy, as they may secure a portion of the reimbursed costs as remuneration. This predictive capacity is enhanced when explicit or implicit `Article 700` claims are identifiable in judgments, underscoring the article's integral role in case outcomes.

This paper aims to achieve the first step: detecting `Article 700` presence in judgments, which relies on predictive models to detect this category of claim. Although the objective seems trivial at first glance through a simple keyword search, our labeled corpora revealed cases of false positives where the expression "article 700" appears, but refers to an earlier claim made in a previous jurisdiction rather than the current one. Moreover, the expression of this claim can vary significantly, depending on the judge's expression and the location in

the judgment i.e., claim section, conclusion section, or reasons section, as can be shown in Table 1 below with three different examples.

Table 1. Examples of variations of the mention `Article 700` in blue.

	In French	**In English**
Example 1	Condamner Mme Exposito à lui verser une somme de 2000 euros de procédure civile, sur le fondement de l'article 700 du code...	Order Mrs. Exposito to pay him the sum of 2000 euros under article 700 of the code...
Example 2	Condamner Mme Exposito à lui verser une somme de 2000 euros des frais de procédure civile.	Order Mrs. Exposito to pay him the sum of 2000 euros in procedural costs.
Example 3	Condamner Mme Exposito à lui verser une somme de 2000 euros d'indemnités procédurales.	Order Mrs. Exposito to pay him the sum of 2000 euros in procedural indemnities.

Court clerks have the flexibility to write and abbreviate mentions in various ways such as "art 700 CPC", signifying "article 700 du code de procédure civile", or "a r t i c l e 700". These variations pose challenges for a simplistic regular expression-based approach. Hence, a comprehensive methodology that can effectively detect demands across all categories is necessary, regardless of whether it relies on specific keywords. Given these complexities and nuances, a simple regular expression is insufficient to address the diverse range of cases encountered. As discussed in Appendix C, an experiment is carried out using a rule-based model (regex) to highlight its limitations in generalization and contextual understanding, as it relies solely on fixed keywords, even for the simplest claim category such as `Article 700`. This highlights the necessity of employing machine learning and deep learning models.

Finally, it is worth mentioning that divergence exists between models designed to predict outcome polarity (acceptation/rejection by the judge) from legal judgment text [9,10] and deep learning models identifying the presence (or not) of specific claim categories (`Article 700` in our case), see e.g., [11]. While both tasks are text classification, their objectives are not the same. Models predicting outcome polarity aim at understanding the overall sentiment expressed in a given text, categorizing it as positive or negative, and therefore providing a probability distribution for the outcome (accept/reject). Conversely, models identifying claim categories concentrate on recognizing the existence of particular predefined claims within the text or not, without necessarily understanding the overall sentiment. In the case of the `Article 700`, we detect whether the plaintiff wins for some given claims but not for all claims, therefore this provides partial information on the outcome, which could be used for outcome prediction. In this work, the focus is only put on the identification of the `Article 700` in French legal judgments.

3. Methodology

3.1. Judicial CamemBERT

BERT [7] is a multi-layer bidirectional transformer encoder that can be fine-tuned on many tasks such as question answering, text classification, named entity recognition, etc. Pretraining a transformer-based model involves solving a masked language modeling (MLM) task: a random subset of the input sequence is masked, and the model is trained to predict the missing elements (tokens).

CamemBERT [12] is a transformer model, derived from RoBERTa [13] and pretrained specifically on French documents. Relying on the attention mechanism, transformers have been shown to achieve state-of-the-art performance on a wide variety of NLP tasks. CamemBERT consists of a stack of transformer encoder layers that sequentially process the input sequence. Each transformer encoder layer is divided into two sublayers: a multi-head self-attention mechanism and a fully connected feedforward network. More precisely, it has 12 transformer encoder layers, each with 12 attention heads and a hidden size of 768.

The Judicial CamemBERT (JC) [14] is a *transformer* model derived from CamemBERT. It was trained on a wide volume of data relating to court decisions, laws and legislation, parliamentary debates, and questions to the government (see Figure 1). JC has been trained at the document level rather than at the sentence level to increase the context length. It was trained on a Masked Language Modeling task (MLM) warm started from the existing CamemBERT checkpoint, which is a general-purpose French language model. Since the length of legal documents exceeds 512 tokens, they are split with an overlap of 32 (*tokens*) so that the model can process them. The architecture is identical to that of CamemBERT, namely 12 layers, 12 attention heads, and a hidden size of 768. The vocabulary is also unchanged and fixed to 32,005 tokens. For the training phase, we rely on batches of 2048 sentences, a dropout (*dropout*) of 0.1, and a linear decreasing learning rate starting from 1e-4. The initial BPC (*Bits Per Character*) and MLM accuracies are 5.079 and 0.518 before learning and 0.741 and 0.874 after training.

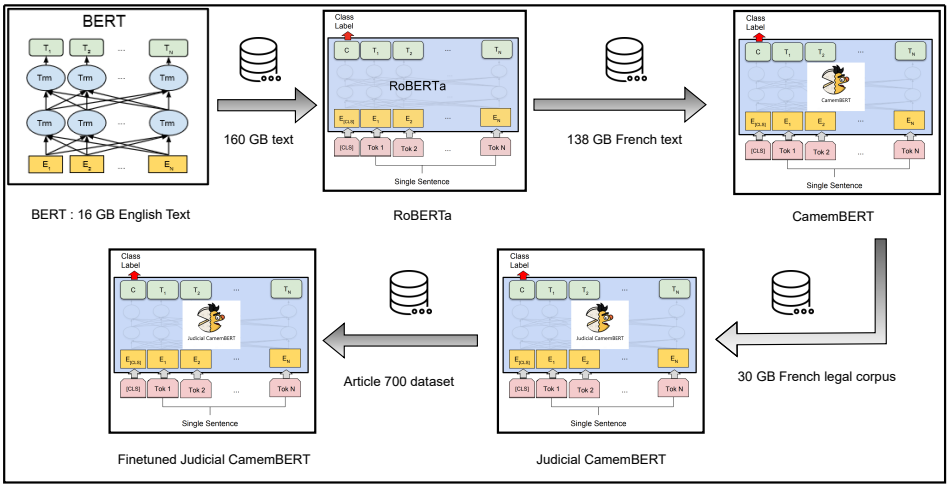

Figure 1. Flowchart showing the specialized Judicial CamemBERT pretraining and finetuning steps. Figure partly made from that of BERT [7].

In our experiment, the Local Sparse Global (LSG [15]) attention is used to support input sequences up to 4096 tokens. Since court decisions are long documents, the LSG attention mechanism allows the model to extrapolate to longer sequences. This provides better contextualization and results in better performances on different tasks such as text summarization, classification, and question answering. The self-attention mechanism allows the model to focus on relevant parts of the input text and ignores irrelevant parts. In the context of judicial documents, irrelevant parts typically refer to sections that do not directly contribute to the classification task or the core decision-making process. These might include particular sections of the document (see Section 4.1), legal citations for other categories, or boilerplate language that repeats across multiple judgments but does not contain information relevant to the specific claims or outcomes. For example, introductory remarks, descriptions of parties, or extensive legal references may not provide significant insight into the actual classification tasks. The self-attention mechanism in transformers helps the model focus on the most salient parts of the document such as the CLAIMS, REASONS, and CONCLUSION (see Section 4.1) while downplaying or ignoring these less pertinent sections.

The attention scores are computed using multiple attention heads, allowing the model to capture different dependencies between the input tokens. In addition, the LSG attention takes sparse blocks of tokens to compress the attention matrix, which reduces computation overhead without sacrificing performance.

Our experiment is based on a fine-tuned Judicial CamemBERT for text classification, specializing the model on a small dataset of labeled text documents. During training, the model's pretrained weights are slightly modified while the task-specific weights of the last

layer are learned from scratch. The input is a sequence of tokens, and the output is a vector of probabilities, one for each possible label. The fine-tuning process involves minimizing a loss function (cross-entropy) through gradient descent, between the predicted probabilities and the true labels.

The main classification task carried out in this article relies on the presence (or absence) of the reference to `Article 700` in the judgment as a whole (or in certain sections of it). `Article 700` is a legal norm that allows any party, whether plaintiff or defendant, to claim reimbursement of legal costs following a court litigation. Although the recourse to `Article 700` is automatic, it does not necessarily appear in judgments (by omission). In general, the claim appears in different forms in the body of the judgment, as shown in Table 1.

To detect the presence of the `Article 700`, the JC is fine-tuned on a simple binary classification task. Since the `Article 700` is often invoked with some terms such as DAMAGES and INTERESTS, the improved contextualization captured by the JC should provide better results compared with simple methods based on frequencies.

3.2. TF-IDF Vectorization and Preprocessing of Judgments

The vectorization TF-IDF, *Term Frequency–Inverse Document Frequency*, has been proposed by [6]. It consists in representing each document (the judgment) as a TF-IDF vector, in which each element of the vector is a score associated with each word of the judgment. The score is decomposed into two components:

- $tf_{w,j}$ is the frequency of word w in judgment j denoted by $f_{w,j}$. It can be represented by the simple frequency (number of words in proportion of all words in j), it can be also represented by a boolean (the word is present in j or not), or other variants such as the normalized logarithm frequency $\ln(1 + f_{w,j})$.
- $idf_{w,j}$ is the inverse document frequency, that is, the inverse proportion of judgments containing word w in the corpus C composed of all judgments j:

$$idf_{w,j} = \ln \frac{|C|}{|\{j : w \in j\}|}$$

In this respect, the TF-IDF representation of word w in a corpus of judgments j is:

$$tfidf_{w,j} = tf_{w,j} \cdot idf_{w,j}$$

Recently some variants of the TF-IDF vectorization has been proposed by [16], in which different weighting schemes are proposed to improve the quality of the TF-IDF vectors.

The detection of `Article 700` in judgments may be performed with machine learning models trained on annotated data provided by legal experts. Then, the annotated data are composed of many sections of the judgment, then the model may be trained over the entire judgment or on particular sections (see Section 4 below). All annotated judgments or sections are preprocessed before the TF-IDF vectorization. The following basic treatments are performed:

- Lower case: All characters are lowercased.
- Stop words: Punctuation marks and recurrent (french) words (le, la, à . . . etc) are removed.
- Numbers are converted in to letters: `Article 700` becomes 'article seven hundreds'.
- Tokenization: The text is segmented into fundamental units, referred to as tokens, to facilitate more efficient processing by the model.
- Lemmatization: All words are lemmatized; for instance, 'gives', 'gave', and 'giving' are reduced to 'give'.

Following these steps, a frequency-based TF-IDF matrix is generated from the preprocessed text, enabling the models to classify judgments based on the extracted input frequency features.

4. Experiments

In this section, data are first presented, then the results of the models based on TF-IDF vectorization (baseline) is analyzed. Finally the results of the fined-tuned Judicial CamemBERT is provided.

4.1. Data

The dataset is composed of 1994 judgments of the French Court of Appeals (see Table 2), which have been annotated by two lawyers. It contains 50% of judgments, which refer to the `Article 700`, and 40.5% of the `Article 700` claims have been accepted by the court. The two legal experts have labeled the judgments (they extract information from judgments). Precisely, 4 sections have been extracted by the legal experts: the HEADER, the REASONS, the CONCLUSION, and the CLAIMS see Table 3 for an example. The header section is easily extracted by regular expression.

- HEADER: It is the first section of the judgment document, where general information about the delivery court, chamber, decision date, jury, parties' names, lawyers, and decision identifier is mentioned. This part of the document never contains the `Article 700`. It is then omitted in the following experimentation.
- CLAIM: The claim section is divided into blocks corresponding to the number of parties involved in the judgment, with each block containing the claims of a specific party. A legal expert selects sentences that express claims relevant to the designated category. The claim is considered absent from the judgment if no such sentences are identified. Accordingly, a binary classifier can be trained over these annotations to discover the specific features (words) associated (or not) to the claim `Article 700`.
- REASONS: In this section of the judgment, the judge explains the reasons for the outcome of the litigation.
- CONCLUSION: In this section of the judgment, the judge makes a little conclusion of the outcome of the litigation, in which the judge provides compensation either to the plaintiff or to the defendant.
- JUDGMENT: It corresponds to the whole judicial document including the four previous sections.

For each judgment, the legal experts can simply indicate whether the `Article 700` is present in the body of the judgment (annotation '1') or not (annotation '0'). Of course, without a specific section of the text related to the claim, it is more difficult for some machine learning models to predict the presence of the claim in the judgment.

Table 2. Descriptive statistics of the dataset (Avg. len. = Average length).

Features	Count	Avg. Len.	Avg. 'Article 700' Frequency
Text of the decision (JUDGMENT)	1956	1556.3	0.001
Text of CLAIM	1956	12.6	0.020
Text of REASONS	1490	16.3	0.013
Text of CONCLUSION	1410	14.9	0.014

Table 2 provides statistics on the training corpus, calculated after the preprocessing steps, including tokenization and lemmatization. The average text length is measured in tokens and is computed as

$$\frac{nb\_tokens\_text}{nb\_texts}$$

The frequency of mentions of 'Article 700' is calculated as

$$\frac{nb\_mentions\_in\_text}{nb\_tokens\_text}$$

for each text. The overall average frequency is then determined by

$$\frac{total\_frequencies\_of\_texts}{nb\_total\_texts}.$$

Table 3 outlines some examples of the 4 sections extracted by the legal experts. By identifying sections of the document upstream with the help of annotators, predictive models can more easily predict the presence (or absence) of `Article 700` and its implicit references. Consequently, it is shown in the following lines the results of different models that have been trained on the whole judgment or simply on small sections that usually contain the mention `Article 700`.

Table 3. Arrêt n°du 13/12/2017 RG n°17/00446 COUR D'APPEL DE REIMS CHAMBRE SOCIALE - Arrêt du 13 Décembre 2017.

	Jugement (in French)	**Judgment (in English)**
HEADER	REPUBLIQUE FRANCAISE aux parties le: AU NOM DU PEUPLE FRANCAIS COUR D'APPEL DE PARIS 4ème Chambre—Section A ARRET DU 10 SEPTEMBRE 2008 (n°, 10 pages) Numéro d'inscription au répertoire général: 07/06621 Décision déférée à la Cour: Jugement du 26 Février 2007—Tribunal de Grande Instance de PARIS-RG n° 04/16946 APPELANTE S.A. . . . agissant poursuites et diligences de son représentant légal représentée par la SCP . . . , avoués à la Cour assistée de Me AL, avocat au barreau de PARIS, INTIMES Madame E. . . A. . . représentée par la SCP . . . , avoués à la Cour assistée de Me DJ, avocat au barreau de PARIS, Monsieur B. . . D. . . représenté par la SCP . . . , avoués à la Cour assisté de Me AN, avocat au barreau de PARIS, plaidant pour . . . COMPOSITION DE LA COUR L'affaire a été débattue le 3 Juin 2008, en audience publique, devant la Cour composée de [. . .] GREFFIER lors des débats Mme F. . . C. . .	REPUBLIQUE FRANCAISE to the parties: ON BEHALF OF THE FRENCH PEOPLE COURT OF APPEAL OF PARIS 4th Chamber—Section A JUDGMENT OF 10 SEPTEMBER 2008 (no., 10 pages) Registration number in the general repertoire: 07/06621 Decision referred to the Court: Judgment of 26 February 2007—Tribunal de Grande Instance de PARIS-RG n°04/16946 APPELLANT S.A. . . . acting in the name of its legal representative represented by SCP . . . , lawyers at the Court assisted by Me AL, lawyer at the PARIS bar, INTIMATE Mrs E. . . A. . . represented by SCP . . . , lawyers at the Court, assisted by Mr DJ, lawyer at the PARIS bar, Mr B. . . D. . . represented by SCP . . . , lawyers at the Court assisted by Mr AN, lawyer at the PARIS bar, pleading for . . . COMPOSITION OF THE COURT The case was debated on 3 June 2008, in public hearing, before the Court composed of [. . .] REGISTRAR during the debates Mrs. F. . . C. . .
REASONS	Le jugement doit être confirmé en ce qu'il a condamné le G. . . aux dépens et en ce qu'il l'a condamné à payer aux défendeurs la somme de 1000 euros au titre de l'article 700 du code de procédure civile.	The judgment must be confirmed insofar as it condemned G. . . to pay the costs and insofar as it condemned him to pay the defendants the sum of 1000 euros under Article 700 of the Code of Civil Procedure.
CONCLUSION	Condamne le G. . . à payer à Madame O.E. veuve L., T.G. L., Madame K.N. épouse L., Madame R. L. veuve S., T.A.S., Madame B.S. et Madame H.S. la somme de 1000 euros au titre de leurs frais irrépétibles d'appel.	Condemns the G. . . to pay to Mrs O.E. widow L., T.G. L., Mrs K.N. wife L., Mrs R. L. widow S., T.A.S., Mrs B.S. and Mrs H.S. the sum of 1000 euros for their unrecoverable appeal costs.
CLAIM	sauf du chef de la demande au titre de l'article 700 du code de procédure civile, portée à la somme de 5000 euros	except for the claim under Article 700 of the Code of Civil Procedure, which is increased to the sum of 5000 euros

To get a relevant annotated label related to `Article 700`, a random sample of 100 annotations is selected to compute an integument score between the two annotators. This sample includes 25 examples from each of the following sections: CLAIM, REASONS, CONCLUSION, and the overall JUDGMENT. The inter-annotator agreement between the two annotators is assessed using the kappa statistic κ, a widely used metric for measuring agreement beyond chance.

The kappa score [17] is calculated using the following formula:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

where P_o is the observed agreement, i.e., the proportion of cases where both annotators make the same annotation; and P_e the expected agreement, i.e., the proportion of cases where agreement is expected by chance. This is calculated based on the marginal probabilities of each annotator's annotations. The kappa statistic ranges from -1 to 1:

1. A kappa value of 1 indicates perfect agreement.
2. A kappa value of 0 indicates agreement equivalent to random chance.
3. A kappa value below 0 suggests that the agreement is worse than random chance.

For this study, the kappa score is found to be 1, indicating complete agreement between the annotators. This result can be attributed to the clarity of the legal article 700 category, which makes it relatively straightforward for human annotators to detect and categorize these annotations with a high degree of consistency.

It is worth noting that such a high level of agreement, while rare in many natural language processing (NLP) tasks, can occur in cases where the task is well-defined, and the categories are unambiguous. The clarity of the text and the well-established nature of the legal categories (such as `Article 700`) likely contributed to the strong inter-annotator reliability observed in this case.

4.2. Baseline Models

Ten binary classifiers are employed to test for the presence (or not) of `Article 700` on the preprocessed judgments, which are vectorized either into unigrams, bigrams, or trigrams. Using the `sklearn` library of Python, 10 classifiers are trained: The logistic regression (logistic with ℓ_1 penalty) [18], SVM (linear and gamma) [19], decision trees (Tree with maximum depth = 5) [20], random forest (RF) [21], multi-layer perceptron (MLP with 1000 iterations) [18], Adaboost, the Gaussian naïve Bayes model (NB) [10], the quadratic discriminant analysis (QDA), the linear discriminant analysis (LDA), the ensemble learning [22] (with soft and hard voting [23]). Recent works also used neural networks for legal judgment prediction [24].

The F-measure gauges the performance of the classifiers. It represents a harmonic mean of the precision (P) and the recall (R). P is a measure of the exactness of classification models. A high precision score indicates a low rate of false positives. R represents the sensitivity of the model to false negative instances. A high recall indicates a low rate of false negative predictions. The F-measure is a combination of those two metrics that balance between them to provide a single score to evaluate the classification model. It is measured as follows:

$$\text{F-measure} = \frac{2 * \text{P} * \text{R}}{\text{P} + \text{R}} \tag{1}$$

Each classifier is separately trained over the 3 parts of the judgments (`Reasons`, `Conclusions`, and `Claim`) or the entire judgment. Each machine learning model should provide better results in predicting `Article 700` when trained on little sections of the judgment rather than on the whole judgment. Also, it could also be faster to train the model over short sections while keeping a very high accuracy. In what follows, the accuracy and the F-measure are reported for the best model only. Table 4 reports the results considering a vectorization of each section into unigrams only.

Table 4. Accuracy and F-measure of the binary models on unigrams (train-test split 80/20 1-fold).

Annotations	Accuracy	F-Measure	Model
Reasons	0.969	0.976	Adaboost
Conclusion	**0.982**	**0.986**	MLP
Claim	0.972	0.972	QDA
Judgment	0.879	0.852	Adaboost

The results in Table 4 show that binary models may suffer from overfitting since in some cases the TF-IDF matrix may have a number of columns greater than the rows. For instance, vectorizing over the entire corpus of judgments provides 1653 rows (judgments) and 32,577 columns (unigrams). In this respect, one solution is to compress the TF-IDF matrix with a principal component analysis (PCA), see e.g. [25] for data feature reduction. Compression is performed with either 200 components, 100, 50, 20 or 10 components.

In the following Table 5, we report the results with a compressed TF-IDF matrix containing both unigrams and bigrams. Before compression, we select the best unigrams and bigrams to get a matrix of lower size (from 1653 $\times$ 449,995 to 1653 $\times$ 64,000), then the PCA is performed on this reduced matrix. Table 5 depicts good results for the section CONCLUSION, this means that the `Article 700` can be easily detected by vectorizing this part of the judgment only. In Appendix A, we provide in detail the results for the 10 classifiers (except the ensemble method) over 200, 100, 50, 20, or 10 principal components (unigrams and bigrams) for each section.

Table 5. Accuracies and F-measures of the binary models on unigrams and bigrams (train-test split 80/20 1-fold).

Annotations	Accuracy	F-Measure	Model	# P. Components
REASONS	0.966	0.967	SVM	20
CONCLUSION	**0.982**	**0.985**	SVM	200
CLAIM	0.974	0.974	QDA	20
JUDGMENT	0.888	0.858	MLP	200

The last experiment is conducted with unigrams, bigrams, and trigrams. Table 6 shows that the results are similar to Table 5. Indeed, feeding the CONCLUSION section to a binary model is the best way to reach high accuracy on the presence of the `Article 700`.

Table 6. Accuracy and F-measure of the binary models on unigrams, bigrams and trigrams (train-test split 80/20 1-fold).

Annotations	Accuracy	F-Measure	Model	# P. Components
REASONS	0.966	0.973	Adaboost	100
CONCLUSION	**0.989**	**0.991**	SVM	200
CLAIM	0.972	0.972	QDA	20
JUDGMENT	0.888	0.861	MLP	100

To address the issue of variability in the train-test split, a 5-fold cross-validation is conducted. In this experiment, a novel compression strategy is introduced to enhance model efficiency. Specifically, a one-hot count vectorizer is trained on a dataset containing 40 categories of claims (18,927 judicial judgments). Then, the difference between the one-hot vectorized matrix of documents within the ARTICLE 700 category and those outside of it is computed. The resulting n-grams are sorted in descending order based on their discriminative power. From this, the top k n-grams are selected, representing those with the most significant difference between the ARTICLE 700 category and other claim categories. The hyperparameter k serves to reduce the dimensionality of the vectorized matrix by retaining only the top k columns, resulting in a more compact matrix of features. This method can offer a computational advantage over compressing the TF-IDF matrix with Principal Component Analysis. Subsequently, training and testing sets are built from these top k n-grams to form the feature matrix for further modeling. Tables 7–9 depict the results with 5-fold cross validation. As can be remarked, the CONCLUSION part of the decision is no more the most relevant part of the document to predict the information about the `Article 700`. Indeed, accuracies and F-measures related to the CLAIM section are higher in the three experiments with respectively unigrams, bigrams and trigrams (except the accuracy on REASONS). Finally, the 5-fold validation indicates that predicting `Article 700`

over the entire JUDGMENT is more challenging, with F-measures ranging between 0.64 and 0.69. Appendix B showcases the results of the 10 binary classifiers on the `Article 700` claim category with 5-fold cross validations using unigrams.

Table 7. Accuracy and F-measure of the models on unigrams (5 fold cross validation).

Annotations	Accuracy	F-Measure	Model	Top k n-grams
REASONS	0.959	0.851	Adaboost	100
CONCLUSION	0.94	0.802	MLP	100
CLAIM	**0.978**	**0.978**	MLP	200
JUDGMENT	0.925	0.682	Adaboost	20

Table 8. Accuracy and F-measure of the models on unigrams and bigrams (5 fold cross validation).

Annotations	Accuracy	F-Measure	Model	Top k n-grams
REASONS	0.953	0.85	SVM	20
CONCLUSION	0.929	0.799	SVM	200
CLAIM	**0.924**	**0.922**	QDA	20
JUDGMENT	0.928	0.69	MLP	200

Table 9. Accuracy and F-measure of the models on unigrams, bigrams and trigrams (5 fold cross validation).

Annotations	Accuracy	F-Measure	Model	Top k n-grams
REASONS	**0.958**	0.847	Adaboost	100
CONCLUSION	0.934	0.812	SVM	200
CLAIM	0.922	**0.921**	QDA	20
JUDGMENT	0.940	0.640	MLP	100

4.3. CamemBERT and JC Experiments

CamemBERT and Judicial CamemBERT [14] are fine-tuned on the binary classification task (presence of `Article 700`). The models are trained as follows: 80% for the training set and 20% for the testing set. Since the JC is warm-started from CamemBERT and trained over a wide corpus of judicial data, better results can be expected. Furthermore, the JC relies on the LSG attention (Local Sparse and Global attention) developed by [15] that allows long sentences of 4096 tokens to be fed into the model. In contrast, the standard CamemBERT can only process up to 512 tokens.

In fact, transformer-based language models have limitations on the maximum input length [26], often measured in tokens rather than characters or words. For example, earlier versions of popular models such as BERT [7] or GPT-2 [27] had a token limit of 512 (base versions) or 1024 tokens (large versions), which constrained the length of text that could be processed at once. However, in this work, we employ a model capable of handling up to 4096 tokens, enabling it to process significantly longer documents without truncation. This expanded token capacity is important for tasks involving legal texts (the whole judgment is a long document), ensuring that essential context is not lost.

Compared with binary models (see Tables 4–6), both CamemBERT models (see the first two columns of Table 10), trained on specific sections of the judgment, *i.e.*, CLAIM, REASONS, or CONCLUSION, yield better results than the binary models, with the highest accuracy of 0.997 observed in the CLAIM section. This is likely due to the fine-tuning of the model on legal data. In contrast, binary models achieved their best performance in the CONCLUSION section, with an F-measure of 0.991 (SVM with unigrams, bigrams, and trigrams). However, when both models are trained on the entire judgment, their accuracies and F-measures decrease, though the JC and CamemBERT base models consistently outperform the binary models. This decline may be due to the length and complexity of judgments, which consist of multiple sections, making it difficult for models to identify distinctive features for

classification. The results are generally lower for both binary and transformer models when trained on the entire judgment using 5-fold cross-validation experiments (see Tables 7–9, and the third column of Table 10), particularly in terms of F-measure, except for the CLAIM section where acceptable to good results are maintained. It is important to note that these results assume the manual extraction of the CLAIM section before model training. In practice, automatically segmenting judgments is a complex task, especially in large corpora. Given the time and cost associated with manual annotation, it is often necessary to rely on the entire judgment for classification. Therefore, despite the challenges, the JC model remains a viable option, as demonstrated by the results obtained on the full JUDGMENT texts.

As shown in Table 10, the JC outperforms CamemBERT. The most important gap relies on detecting the `Article 700` on the whole judgment, a gap of 3% of absolute points between the F-measures. Indeed, in this case, JC takes advantage of its self-attention based on the LSG architecture.

Table 10. CamemBERT results related to the presence of the `Article 700` (F1 macro).

Annotations	CamemBERT Base		Judicial CamemBERT		JC (5-Fold Cross Validation)	
	Accuracy	F-Measure	Accuracy	F-Measure	Accuracy	F-Measure
REASONS	0.935	0.884	0.945	0.901	**0.963**	0.874
CONCLUSION	0.94	0.898	**0.943**	0.908	0.929	0.79
CLAIM	0.985	0.985	**0.997**	**0.997**	0.992	0.992
JUDGMENT	0.934	0.839	**0.951**	0.869	0.933	0.658

The cross-validation shows that, as mentioned with the results of Tables 4–6, the CLAIM part of the judgment enables to reach 0.99 accuracy and F-measure. Taking either the whole JUDGMENT or the CONCLUSION section does not allow the binary model to detect the presence or the absence of ARTICLE 700 which can explain a lower accuracy and F-measures outlined in Table 10 on 5-fold cross validation.

To show that our models can be generalized to other datasets related to new claims different from the ARTICLE 700 category, we provide an new experiment on 13 claim categories. Since the JC outperforms CamemBERT base, we present only the results of JC to outline a comparison with the binary models. A description of the other claim categories is provided in Appendix D.

As Table 11 outlines, traditional machine learning models based on the TF-IDF vectorization may outperform the Judicial CamemBERT. This experiment has been conducted over 13 claim categories by taking into account the whole judgment. A 5-fold validation shows that on 13 categories binary models provide better results. To be precise, 3 categories are detected with 1.0 accuracy (and F1 score) by both binary classification and Judicial CamemBERT models. Moreover, traditional binary models beat the Judicial CamemBERT, which is specialized in law, in 8 categories out of 13.

Table 11. Performance of Best Binary Models and JC Model on Different Categories (5 fold cross validation training on the whole judgment).

Category	Binary Model	Binary Model Accuracy (5-fold)	Binary Model F1 (5-fold)	JC Accuracy (5-fold)	JC F1 (5-fold)	Dataset Size
NFA-33A	LR	1.0	1.0	1.0	1.0	1240
NFA-22D	Gaussian NB	0.982	0.983	0.973	0.973	1346
NAC-80C	AdaBoost	0.897	0.938	0.922	0.955	1028
NAC-80A-B	LR	1.0	1.0	1.0	1.0	3210
NAC-80A-A	SVM	1.0	1.0	0.990	0.990	1216
NAC-64B-B	SVM	0.986	0.986	0.989	0.989	2184

Table 11. *Cont.*

Category	Binary Model	Binary Model Accuracy (5-fold)	Binary Model F1 (5-fold)	JC Accuracy (5-fold)	JC F1 (5-fold)	Dataset Size
NAC-64B-A	SVM	0.969	0.970	0.829	0.858	2066
NAC-64A-B	RF	0.879	0.882	0.707	0.622	1026
NAC-64A-A	RF	0.988	0.989	0.977	0.976	1004
NAC-59A-C	DT	0.990	0.990	0.980	0.981	1180
NAC-59A-B	LDA	0.984	0.984	0.955	0.955	2218
NAC-59A-A	RF	0.995	0.995	0.990	0.990	1168
NAC-14A	LDA	0.994	0.994	0.968	0.970	932

5. Conclusions

In this study, it was shown that the detection of the `Article 700` may be done with both traditional machine learning models trained on a matrix of term frequencies and inverse document frequencies and transformer models such as CamemBERT and Judicial CamemBERT.

The transformer models benefit from their multi-head attention mechanisms that capture the context of the document with long sentences (LSG attention). The machine learning models reach competitive results compared to CamemBERT Base, especially for the conclusion section and judgment, as the 5-fold cross-validation results demonstrate. Such results can be explained by the usage of frequent words vector, which captures the category relevant keywords. Our Judicial CamemBERT reaches 0.951 accuracy on the whole judgment or 0.997 accuracy on the annotated sentences that contain the different expressions of the `Article 700` claims. However, this result may be attenuated since the 5-fold cross-validation shows an F-measure of 0.658, which highlights the difficulty of the task on the whole judgment to capture relevant words in a long complex context. This can be explained by the complexity of the language and the length of the decisions being part of the training dataset. Certain cases occur with lower frequency, where the `Article 700` is expressed differently, due to the dataset splits, it can be non-encountered during training which leads to such low performance.

We also can notice that the CLAIM section is the most reliable and deterministic in all configurations, which can be concluded from the general best metrics with TF-IDF-based models or models based on Transformers.

Our approach can be generalized to other claim categories, as demonstrated by our 5-fold cross-validation experiments across 13 categories, which show consistently high performance in overall judgment accuracy. However, certain categories, such as `NAC-64A-B`, may be more challenging to classify. This difficulty can arise from the absence of distinctive keywords or the presence of overlapping keywords with other categories, which can reduce the accuracy of classification models.

Both models demonstrate high performance; however, basic machine learning models are more cost-effective in terms of computational resources usage. This indicates that traditional binary models can still be highly efficient for classification tasks. Transformer-based models, on the other hand, require domain-specific knowledge to achieve robustness in complex text classification tasks, particularly in the legal domain.

Future research could focus on outcome prediction, a challenging task that involves identifying the anonymized parties, each party's claims, and the final outcomes of those claims. Another avenue for future work could involve explaining judicial decisions by identifying the words and expressions most significant to the classifiers. Explainable AI (XAI) models, such as those based on Shapley values [28], could be utilized to assign importance scores to the expressions influencing judges' decisions, particularly with respect to `Article 700` or other relevant categories.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the Agence Nationale de la Recherche for funding the project LAWBOT ANR-20-CE38-0013. Sid Ali Mahmoudi acknowledges a doctoral grant from the Occitanie region.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data can be found at https://lawbot.unimes.fr/data-sets/, accessed on 13 November 2024.

Conflicts of Interest: The authors declare no conflict of interest. The funder had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Comparisons of the Binary Classifiers (Unigrams and Bigrams with PCA Compression)

Table A1. Comparison of binary models: train-test split 80/20 1-fold #200 components on CONCLUSION.

Model (`sklearn` Python Specification)	Accuracy	F-Measure
LogisticRegression(penalty='l1', solver='saga', tol=0.01)	0.9645	0.9711
SVM: SVC(C=1, gamma=2)	**0.9823**	0.9857
DecisionTreeClassifier(max_depth=5)	0.9468	0.9573
RandomForestClassifier()	0.9716	0.9773
MLPClassifier(max_iter=1000)	0.9716	0.9771
AdaBoostClassifier()	0.9539	0.9628
GaussianNB()	0.9255	0.9395
QuadraticDiscriminantAnalysis()	0.9539	0.9634
LinearDiscriminantAnalysis()	0.9752	0.9798

Table A2. Comparison of binary models: train-test split 80/20 1-fold #20 components on REASONS.

Model (`sklearn` Python Specification)	Accuracy	F-Measure
LogisticRegression(penalty='l1', solver='saga', tol=0.01)	0.9497	0.9577
SVM: SVC(C=1, gamma=2)	**0.9664**	0.9727
DecisionTreeClassifier(max_depth=5)	0.9094	0.9244
RandomForestClassifier()	0.9396	0.9500
MLPClassifier(max_iter=1000)	0.9497	0.9589
AdaBoostClassifier()	0.9497	0.9584
GaussianNB()	0.9228	0.9373
QuadraticDiscriminantAnalysis()	0.9463	0.9563
LinearDiscriminantAnalysis()	0.9530	0.9607

Table A3. Comparison of binary models: train-test split 80/20 1-fold #200 components on CLAIM.

Model (`sklearn` Python Specification)	Accuracy	F-Measure
LogisticRegression(penalty='l1', solver='saga', tol=0.01)	0.9643	0.9630
SVC(C=1, gamma=2)	0.9643	0.9628
DecisionTreeClassifier(max_depth=5)	0.9617	0.9600
RandomForestClassifier()	0.9719	0.9710
MLPClassifier(max_iter=1000)	0.9694	0.9684
AdaBoostClassifier()	0.9643	0.9628
GaussianNB()	0.9388	0.9381
QDA: QuadraticDiscriminantAnalysis()	**0.9745**	0.9741
LinearDiscriminantAnalysis()	0.9541	0.9519

Table A4. Comparison of binary models: train-test split 80/20 1-fold #200 components on JUDGMENT.

Model (`sklearn` Python Specification)	Accuracy	F-Measure
LogisticRegression(penalty='l1', solver='saga', tol=0.01)	0.7372	0.6201
SVC(C=1, gamma=2)	0.8580	0.8112
DecisionTreeClassifier(max_depth=5)	0.8369	0.7891
RandomForestClassifier()	0.8580	0.8142
MLP: MLPClassifier(max_iter=1000)	**0.8882**	0.8582
AdaBoostClassifier()	0.8792	0.8561
GaussianNB()	0.7946	0.7622
QuadraticDiscriminantAnalysis()	0.6314	0.5933
LinearDiscriminantAnalysis()	0.8731	0.8456

Appendix B. Comparisons of the Binary Classifiers (Unigrams on 5-Fold Cross Validation)

Table A5. Binary models results: 5-fold cross-validation on the CLAIMS dataset.

Model (`sklearn` Python Specification)	Accuracy	F-Measure
LogisticRegression(penalty='l1', solver='saga', tol=0.01)	0.9740	0.9737
SVC(C=1, gamma=2)	0.9705	0.9701
DecisionTreeClassifier(max_depth=5)	0.9688	0.9685
RandomForestClassifier()	0.9745	0.9746
GaussianNB()	0.9543	0.9537
MLPClassifier(max_iter=1000)	**0.9780**	**0.9780**
AdaBoostClassifier()	0.9716	0.9714
QuadraticDiscriminantAnalysis()	0.8842	0.8756
LinearDiscriminantAnalysis()	0.9716	0.9712
Ensemble method Vote HARD	0.9757	0.9755
Ensemble method Vote SOFT	0.9757	0.9756

Table A6. Binary models results: 5-fold cross-validation on the REASONS dataset.

Model (`sklearn` Python Specification)	Accuracy	F-Measure
LogisticRegression(penalty='l1', solver='saga', tol=0.01)	0.9519	0.8442
SVC(C=1, gamma=2)	0.9553	0.8536
DecisionTreeClassifier(max_depth=5)	0.9567	0.8572
RandomForestClassifier()	0.9404	0.8137
GaussianNB()	0.9269	0.7737
MLPClassifier(max_iter=1000)	0.9560	0.8405
AdaBoostClassifier()	**0.9590**	0.8510
QuadraticDiscriminantAnalysis()	0.9086	0.7470
LinearDiscriminantAnalysis()	0.9526	0.8187
Ensemble method Vote HARD	0.9587	0.8612
Ensemble method Vote SOFT	0.9534	0.8639

Table A7. Binary models results: 5-fold cross-validation on the CONCLUSION dataset.

Model (`sklearn` python specification)	Accuracy	F-Measure
LogisticRegression(penalty='l1', solver='saga', tol=0.01)	0.9323	0.8066
SVC(C=1, gamma=2)	0.9209	0.8013
DecisionTreeClassifier(max_depth=5)	0.9288	0.7944
RandomForestClassifier()	0.9196	0.7770
GaussianNB()	0.9104	0.7314
MLPClassifier(max_iter=1000)	**0.9401**	0.8020
AdaBoostClassifier()	0.9357	0.8222
QuadraticDiscriminantAnalysis()	0.8349	0.6246
LinearDiscriminantAnalysis()	0.9358	0.7874
Ensemble method Vote HARD	0.9373	0.8106
Ensemble method Vote SOFT	0.9315	0.8251

Table A8. Binary models results: 5-fold cross-validation on the JUDGMENT dataset.

Model (`sklearn` python specification)	Accuracy	F-Measure
LogisticRegression(penalty='l1', solver='saga', tol=0.01)	0.9249	0.6738
SVC(C=1, gamma=2)	0.9203	0.6700
DecisionTreeClassifier(max_depth=5)	0.9249	0.6717
RandomForestClassifier()	0.9117	0.6411
GaussianNB()	0.8590	0.5390
MLPClassifier(max_iter=1000)	0.9366	0.5977
AdaBoostClassifier()	**0.9250**	0.6820
QuadraticDiscriminantAnalysis()	0.8523	0.5190
LinearDiscriminantAnalysis()	0.9366	0.6936
Ensemble method Vote HARD	0.9285	0.6883
Ensemble method Vote SOFT	0.9346	0.6780

Appendix C. Rule-Based Model Results on the ARTICLE 700 Claim Category

Table A9. Rule-based model results across sections (5-fold cross-validation).

Section	Accuracy	F-Measure
CLAIMS	0.9311	0.9305
REASONS	0.9206	0.7974
CONCLUSION	0.9115	0.7816
JUDGMENT	0.3024	0.1917

Appendix D. Claim Categories Identifiers

NAC-14A	violation of privacy
NAC-59A-A	nullity of contract due to lack of consent
NAC-59A-B	void contract for defective consent: mistake
NAC-59A-C	void contract for defective consent: violence
NAC-64A-A	qualification of abnormal neighborhood disturbance
NAC-64A-B	injunction for abnormal neighborhood disturbance
NAC-64B-A	damages for abuse of the right to sue
NAC-64B-B	classification as unfair competition
NAC-80A-A	nullity of discriminatory job termination
NAC-80A-B	damages for unfair job termination
NAC-80C	damages for moral harassment of an employee
NFA-22D	violation of the adversarial principle
NFA-33A	prescription denied

References

1. Direction des Affaires Civiles (Ed.). Statistique sur la Profession d'Avocat—Situation au 1er Janvier 2020. 2021. Available online: https://www.justice.gouv.fr/sites/default/files/migrations/portail/art_pix/statistique_sur_la%20profession_avocat_2020.pdf (accessed on 20 November 2024).
2. Condevaux, C. Neural Legal Outcome Prediction with Partial Least Squares Compression. *Stats* **2020**, *3*, 396–411. [CrossRef]
3. Medvedeva, M.; Vols, M.; Wieling, M. Using machine learning to predict decisions of the European Court of Human Rights. *Artif. Intell. Law* **2020**, *28*, 237–266. [CrossRef]
4. Mathis, B. Extracting Proceedings Data from Court Cases with Machine Learning. *Stats* **2022**, *5*, 1305–1320. [CrossRef]
5. Vuong, Y.T.H.; Bui, Q.M.; Nguyen, H.T.; Nguyen, T.T.T.; Tran, V.; Phan, X.H.; Satoh, K.; Nguyen, L.M. SM-BERT-CR: A deep learning approach for case law retrieval with supporting model. *Artif. Intell. Law* **2022**, *31*, 601–628. [CrossRef]
6. Salton, G.; Buckley, C. Term-weighting Approaches in Automatic Text Retrieval. *Inf. Process. Manag.* **1988**, *469*, 513–523. [CrossRef]
7. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
8. Martin-Serf, A. Frais et dépens. La créance de dépens et frais résultant de l'article 700 du code de procédure civile a son origine dans la décision qui statue sur ces frais et dépens. *Rev. Trimest. Droit Commer. Droit Écon.* **2010**, *1*, 199.
9. Bertalan, V.G.F.; Ruiz, E.E.S. Using attention methods to predict judicial outcomes. *Artif. Intell. Law* **2022**, *32*, 1–29.
10. Shaikh, R.A.; Sahu, T.P.; Anand, V. Predicting outcomes of legal cases based on legal factors using classifiers. *Procedia Comput. Sci.* **2020**, *167*, 2393–2402. [CrossRef]
11. Chalkidis, I.; Fergadiotis, E.; Malakasiotis, P.; Aletras, N.; Androutsopoulos, I. Extreme Multi-Label Legal Text Classification: A Case Study in EU Legislation. In Proceedings of the Natural Legal Language Processing Workshop 2019, Minneapolis, MN, USA, 7 June 2019; pp. 78–87.
12. Martin, L.; Muller, B.; Suárez, P.J.O.; Dupont, Y.; Romary, L.; de La Clergerie, É.V.; Seddah, D.; Sagot, B. CamemBERT: A tasty French language model. *arXiv* **2019**, arXiv:1911.03894.
13. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692,
14. Mahmoudi, S.A.; Condevaux, C.; Mathis, B.; Zambrano, G.; Mussard, S. NER sur décisions judiciaires françaises: CamemBERT Judiciaire ou méthode ensembliste? In Proceedings of the Extraction et Gestion des Connaissances, EGC 2022, Blois, France, 24–28 January 2022; Amer-Yahia, S., Soulet, A., Eds.; RNTI: Paris, France, 2022; Volume E-38, pp. 281–288.
15. Condevaux, C.; Harispe, S. LSG Attention: Extrapolation of Pretrained Transformers to Long Sequences. In Proceedings of the Advances in Knowledge Discovery and Data Mining, Osaka, Japan, 25–28 May 2023; Kashima, H., Ide, T., Peng, W.C., Eds.; Springer: Cham, Switzerland, 2023; pp. 443–454.

16. Ngompé, G.T.; Mussard, S.; Zambrano, G.; Harispe, S.; Montmain, J. Identification of Judicial Outcomes in Judgments: A Generalized Gini-PLS Approach. *Stats* **2020**, *3*, 427–443. [CrossRef]
17. Cohen, J. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **1960**, *20*, 37–46. [CrossRef]
18. Strickson, B.; De La Iglesia, B. Legal judgement prediction for uk courts. In Proceedings of the 3rd International Conference on Information Science and Systems, Cambridge, UK, 19–22 March 2020; pp. 204–209.
19. Aletras, N.; Tsarapatsanis, D.; Preoţiuc-Pietro, D.; Lampos, V. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. *PeerJ Comput. Sci.* **2016**, *2*, e93. [CrossRef]
20. Santosh, T.; Xu, S.; Ichim, O.; Grabmair, M. Deconfounding Legal Judgment Prediction for European Court of Human Rights Cases Towards Better Alignment with Experts. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 1120–1138.
21. Chen, H.; Wu, L.; Chen, J.; Lu, W.; Ding, J. A comparative study of automated legal text classification using random forests and deep learning. *Inf. Process. Manag.* **2022**, *59*, 102798. [CrossRef]
22. Dietterich, T.G. Ensemble methods in machine learning. In Proceedings of the International Workshop on Multiple Classifier Systems, Reykjavik, Iceland, 10-12 June 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 1–15.
23. Zhou, Z.H. *Ensemble Methods: Foundations and Algorithms*; CRC Press: Boca Raton, FL, USA, 2012.
24. Chalkidis, I.; Androutsopoulos, I.; Aletras, N. Neural legal judgment prediction in English. *arXiv* **2019**, arXiv:1906.02059.
25. Shang, X. A computational intelligence model for legal prediction and decision support. *Comput. Intell. Neurosci.* **2022**, *2022*, 5795189. [CrossRef] [PubMed]
26. Dong, Z.; Tang, T.; Li, L.; Zhao, W.X. A survey on long text modeling with transformers. *arXiv* **2023**, arXiv:2302.14502.
27. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
28. Lundberg, S.M.; Lee, S.I. deepSHAP: Explaining Deep Learning Models Using Shapley Values. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 4765–4774.