Proceedings of the Seventeenth International Workshop on Juris-Informatics 2023 (JURISIN 2023)

in association with
the 15th JSAI International Symposia on AI (JSAI-isAI 2023)

JURISIN 2023 Co-chairs

Ken Satoh, National Institute of Informatics, Japan
Katsumi Nitta, National Institute of Informatics, Japan

June 5-6, 2023

Preface

This volume contains 20 papers presented at the 17th Workshop of Juris Informatics, JURISIN 2023, held in Kumamoto, Japan, June 5-6, 2023.

Juris informatics is an interdisciplinary discipline that studies various legal issues from an informatics perspective.

The international workshop on juris informatics, JURISIN, began in 2007 and has been held once a year with the support of the Japanese Society for Artificial Intelligence. Although only nine related topics were exemplified in the first JURISIN call for papers, including legal reasoning, argumentation agents, and legal ontology, in recent years, the development of artificial intelligence technology has greatly expanded the scope of problems to be solved, including the use of machine learning and the legal and social problems caused by artificial intelligence. As a result, topics in 24 fields were given as examples in this year's JURISIN call for papers.

As a result, diverse papers related to artificial intelligence and law were submitted from more than 10 countries. Each paper was reviewed by three reviewers, from which 21 papers were accepted. Among them were important research themes such as representation of legal knowledge, as well as research themes that have been the focus of much attention in recent years, such as machine learning and privacy. In addition to the general presentations, two invited talks will be given by Dr. Satoshi Tojo (Asian University) and Dr. Denis Merigoux (INRIA). This volume contains 20 papers since one paper was withdrawn for a publication in the proceedings.

Finally, we would like to express our deepest gratitude to those who submitted papers, to the PC members who reviewed the papers, and to the Japanese Society for Artificial Intelligence for providing the venue for this workshop.

June 5 and 6, 2023 Ken Satoh
Tokyo Katsumi Nitta

Program Committee

Michał Araszkiewicz	Jagiellonian University
Ryuta Arisaka	Kyoto University
Marina De Vos	University of Bath
Kripabandhu Ghosh	Indian Institute of Science Education and Research (IISER) Kolkata, India
Saptarshi Ghosh	Indian Institute of Technology Kharagpur
Randy Goebel	University of Alberta
Guido Governatori	Independent researcher
Tokuyasu Kakuta	Chuo University
Yoshinobu Kano	Shizuoka University
Mi-Young Kim	Department of Computing Science, U. of Alberta, Canada
Nguyen Le Minh	Graduate School of Information Science, Japan Advanced Institute of Science and Technology
Makoto Nakamura	Niigata Institute of Technology
Katsumi Nitta	National Institute of Advanced Indutrial Science and Technology
Yasuhiro Ogawa	Nagoya City University
Juliano Rabelo	AMII
Seiichiro Sakurai	Meiji Gakuin University
Ken Satoh	National Institute of Informatics and Sokendai, Japan
Akira Shimazu	JAIST
Satoshi Tojo	Asia University
Katsuhiko Toyama	Nagoya University
Bart Verheij	University of Groningen
Yueh-Hsuan Weng	Tohoku University
Masaharu Yoshioka	Hokkaido University
Thomas Ågotnes	University of Bergen

Additional Reviewers

Fungwacharakorn, Wachara
Hayashi, Hisashi
Nguyen, Ha-Thanh
Tsushima, Kanae

Table of Contents

Exploring the Explainability and Legal Implications of Regression Models in Transportation Domain

Gayane Grigoryan[1][0000−0002−8567−9643], Livio Robaldo[2], Cesar Ariel Pinto[1], and Andrew J. Collins[1]

[1] Engineering Management and Systems Engineering Department, Old Dominion University, Norfolk, Virginia, USA
[2] Legal Innovation Lab Wales, Swansea University, UK

Abstract. Artificial Intelligence (AI) applications can be found in various real-world systems, including vehicle system design and real-time car accident prediction. There is an increasing need to better explain AI-driven processes, especially in terms of potential legal disputes that might result from AI decisions. For this reason, more and more Explainable Artificial Intelligence (XAI) methods are under development to explain black-box models. The objective is to improve the models' explainability fit to reduce legal or ethical issues caused by wrongful explanations. This paper discusses use cases in the transportation domain and how the explainability of a regression model goes wrong when there is a lack of clarity in the relationship between the features and the target. The paper then proposes to use an XAI technique called Shapley net effects to improve the explainability of the model. The purpose of this paper is three-fold, namely, to show that XAI is useful for inherently explainable models, to demonstrate the practical use of XAI for a transportation system, and to highlight the legal problems and liabilities generated due to the model misrepresenting the prediction for a transportation system. The data and the implementation is available at https://github.com/grigoryangayane/RegressionShapley.

Keywords: machine learning · explainable AI · legal AI · cooperative game theory · Shapley values · multicollinearity

1 Introduction

AI is being utilized in various domains like in-car design systems, natural language processing for news reporting, and medical decision-support tools. As these applications become entangled in legal disputes [16] [15], it is essential to make AI methods more explainable. As a result, explainable artificial intelligence (XAI) is becoming particularly relevant in LegalTech. The term Responsible AI (RAI) has been coined as a step beyond XAI, and practitioners believe that concepts of responsibility and explainability should be primarily considered from the legal perspective due to the challenges posed by the operation of AI-based

systems on individuals' rights and freedoms. The field of law needs to provide well-defined interpretations of these concepts, and reasoning procedures based on them should be clarified. Legal challenges in the use of AI systems include liability, the (re)interpretation of classical legal concepts, and the distribution of liability among involved actors. The goal is to integrate methodological AI, ethical, and legal perspectives regarding responsibility and accountability.

The US has implemented various laws and regulations aimed at ensuring AI models are explainable, fair, transparent, and used responsibly and ethically. For instance, the Fair Credit Reporting Act (FCRA) mandates that any automated system used in credit evaluation, including AI models, must be transparent and explainable to consumers. The Civil Rights Act of 1964 prohibits discrimination based on factors such as race, religion, national origin, and sex. The General Data Protection Regulation (GDPR), a European Union law, also applies to US companies that process personal data of EU citizens. Furthermore, the California Consumer Privacy Act (CCPA) requires businesses to grant consumers the right to access and delete their personal data, in addition to disclosing the types of data collected and the purposes for which it is used. This legislation also includes provisions for the transparency of AI models used in decision-making and profiling. Other relevant laws and regulations include the Americans with Disabilities Act (ADA), the Electronic Communications Privacy Act (ECPA), and the Children's Online Privacy Protection Act (COPPA).

Ensuring explainability in AI models can be challenging, especially when a model is intuitive and reasonable overall but lacks statistical power due to insufficient data, dependent data, or an important variable being omitted. For example, a machine learning algorithm could misdiagnose a medical condition, a self-driving car could run over pedestrians, or a partially-operated drone could cause crashes [17]. In 2018, a pedestrian was killed by an Uber self-driving car in Arizona. The car's AI system did not explain why it failed to recognize the pedestrian, which raised questions about the safety of self-driving cars. The contributing factors to the system misidentification and misinterpretation were the failure of the Uber operator to monitor the road and the automated system, an inadequate safety culture at Uber, and insufficient regulatory oversight.

In the transportation domain, it is imperative to comprehend the legal and ethical issues of AI applications to address misconduct, prevent discrimination, and ensure transparency. In the legal realm, it is essential to clarify how a particular decision was made to avoid accidents. The legal challenges here are based on regulations and may result in legal action if those rules are violated [14]. On the other hand, ethical challenges demand a more nuanced approach since they are not always governed by specific laws and may not have legal consequences [14]. Nevertheless, ethical considerations play a crucial role in promoting responsible AI practices and maintaining the public's trust in AI systems, given the unique challenges of the transportation industry.

To tackle some of these issues, we present an evaluation of two problems related to vehicle design and accidents using a regression model approach, revealing the vulnerability of the model's outcomes, even when the data and the model

are sound. To mitigate these risks, we employ a hybrid XAI modeling technique, which helps to elucidate the relationship between features and targets. We utilize two datasets - seatpos and US accidents. The former captures associations between car seat positions and anthropometric dimensions of drivers, while the latter provides insights into accident casualties and the impact of environmental factors such as precipitation on accident occurrences.

The next section provides background on explainability and explainable AI (XAI). Then we describe background information about the regression model, followed by the case study. The paper is concluded in final section 6.

2 Background

Explainable artificial intelligence has been gaining increasing attention in the last few years [6]. Explainability in machine learning is generally characterized as the ability of the human user to describe the model's logic. Explainable models deliver content that can be verbal, visual, or written, provide clarification, attempt justification, and establish fidelity and trust [4]. Explainable AI provides strategic value and competitive advantage for businesses, builds trust and confidence of stakeholders in the ML, and helps to determine whether discrimination occurred and respectively identify legal or ethical issues [17].

Explainability is needed for tasks that require compliance with law, as the application of the law must, by definition, be fair, transparent, and bias-free. Specifically, explainability could be essential to describe and clarify situations and settings associated with critical decisions, such as determining the product liability of defects in vehicles, where these defects can be correlated with traffic accidents. Features, such as the vehicle color, may have a significant effect on the traffic accident [21]. Accidents may also occur as the result of a faulty interior design of a vehicle such as doors, seats, cushions, knobs, steering wheels, or the overhead structure [20]. In our paper, we discuss a car seat design problem given the different anthropometric characteristics of a driver, where these characteristics are highly correlated with each other. For the next data, we analyze the relationship between weather conditions and car accidents. This information is crucial for both manufacturers and customers as it can aid in the prevention of accidents.

There are multiple approaches to improve explanations in AI. One approach is using logical reasoning based on logical formulas and symbolic representations. However, pure logical approaches based on formulas and ontologies can be time-consuming to build and update, and symbolic knowledge may become outdated once the analyzed field changes. On the other hand, machine learning is used to learn patterns, detect anomalies, and make predictions automatically. Another way to achieve explainability is to use inherently explainable models, such as decision trees or regression models [18], or apply post-hoc explanations to the model. Post-hoc explanations suggest a separate set of techniques to explain highly complex uninterpretable models with high accuracy [10].

In this paper, we utilize a regression model, which is inherently explainable, as it is not computationally intensive, and its model construct and results are easy to interpret. However, a linear model may have limitations when applied to complex problems, particularly if some classical linear regression model assumptions are violated, which could result in biased outcomes and an inaccurate explanation. In such cases, where the linear model is still appropriate but an assumption is violated, post-hoc explainable model techniques can be useful in clarifying the relationship between the features of interest. Our paper aims to demonstrate the use of an XAI technique to provide an explanation of an intrinsically explainable model in scenarios where all assumptions hold (US accident dataset), as well as when some are violated (seatpos dataset).

The following section presents a background of the regression model.

3 Regression model overview

Multiple linear regression models can handle numerous features to elucidate a greater amount of variation in the predicted variable. [7]. The mathematical form of a regression model is as follows:

$$y_i = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n + \varepsilon \tag{1}$$

Where, y_i are the observations of the target variable, x_1, x_2, ..., x_n are the features, ε is the regression error term, that is assumed to be normally distributed, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

The output produced by regression models is considered to be naturally interpretable because it is simple to comprehend the mathematical calculations that led to a specific outcome. Through the computation of partial differential equations, a multiple linear regression model establishes the connection between the features and the target variable.

The regression analysis yields the intercept (β_0) and regression coefficients (β_1, β_2, ..., β_n) that can be obtained using the Ordinary Least Square approach (OLS). Additionally, the regression model provides two statistical measures to assess its performance. The first is the multiple determination R^2 coefficient, which measures how well the model replicates observed outcomes based on the proportion of total variation explained by the model. The second statistic is the p-value, which evaluates the null hypothesis that the regression coefficient is equal to zero. If the coefficient equals 0 or is not statistically significant, then the feature has no effect on the model. A low p-value (< 0.05) suggests that the null hypothesis can be rejected, meaning that a feature with a low p-value is likely to be a valuable addition to the model. Conversely, larger p-values indicate that the features are not suitable for predicting the target variable.

Regression results can be unreliable when produced by a model with highly correlated features. In such cases, the output may show statistically insignificant p-values, indicating that some features are irrelevant and should be removed.

It is necessary to use more effective explanation methods and conduct a thorough analysis to handle such situations. Multicollinearity, which happens when multiple features in a model have a high correlation, is a common problem.

Next, we present case studies of how certain features can have practical significance for the model in real-world scenarios, despite their effect sizes on predicting the target variable being very small as suggested by the regression coefficients or the p-values being insignificant.

4 Case study

There are two case studies considered in this paper. The first case study investigates the relationship between car seat design, the driver's anthropometric characteristics (seatpos dataset), and, the second considers the risk of accidents under various weather conditions (US accident dataset). The first case study is important as interior design has been linked to traffic accidents in previous studies [20]. To address this issue, various measures have been taken to establish better car designs. The insights gained from this study could be invaluable to car manufacturers and policymakers in designing safer vehicles and reducing the incidence of accidents.

The subsection below describes the datasets used for the case study, followed by the results of the regression model described in the previous section.

4.1 Data

The seatpos dataset was collected by HuMoSim laboratory researchers at the University of Michigan, is intended to study a car seat position given several anthropometric parameters describing 38 drivers. A detailed description of the dataset and an example of its use can be found in Faraway [2]. Human body dimensions are described to be symmetric. Features included in the dataset to model the car seat position are age in years (Age), weight in lbs (Weight), height in shoes in cm (HtShoes), height bare foot in cm (Ht), seated height in cm (Seated), lower arm length in cm (Arm), thigh length in cm (Thigh), lower leg length in cm (Leg), a horizontal distance of the midpoint of the hips from a fixed location in the car in mm (hipcenter). Knowing the dimensions of the driver helps the manufacturer in designing a car seat that provides the maximum possible safety. In the regression model, the hipcenter is the target variable and proxy measurement for car seat, and the rest of the variables are the features to explain the hipcenter.

The US accident dataset a countrywide Traffic Accident Dataset (2016 - 2020) is used to examine the impact of weather conditions on accident risk. The dataset contains 2,974,335 observations, and the ordinal variable Severity as the response variable, with a value from 1 to 4 describing the amount of interference the accident had on traffic. In this study, the goal is to determine which factors of car accidents are most associated with resulting traffic interference by analyzing

the other variables in the dataset. For the analysis, we selected a subset of 100,000 features from the large dataset and removed features that were not practically significant for the prediction, such as the index of the feature. The remaining features used in the model include distance, temperature, wind chill, humidity, pressure, visibility, wind speed, and precipitation.

The output of the regression models is presented in the subsection below.

4.2 Results

First, we discuss the two base multiple regression models for the two datasets. The initial models for both datasets are presented in Table 1.

Seatpos: The multiple regression results are insignificant with insignificant p-values. Important questions here are if we should trust the model and how reliably the model explains the relationship between the driver dimensions and the car seat design. However, the model outcome seems to be unsatisfactory. The car seat design requirements suggest that drivers' dimensions play a substantial role when designing a car seat.

Table 1. Initial results of the regression models

Seatpos	Estimate	P_value	US accident	Estimate	P_value
Age	0.77	0.18	Distance	0.1134	2e-16
Weight	0.02	0.93	Temperature	0.008	2e-16
HtShoes	-2.69	0.78	Wind Chill	-0.008	2e-16
Ht	0.6	0.95	Humidity	0.00023	0.00289
Seated	0.53	0.88	Pressure	-0.003931	0.011
Arm	-1.32	0.73	Visibility	0.0008	0.2
Thigh	-1.14	0.67	Wind Speed	-0.0007	0.01
Leg	-6.43	0.18	Precipitation	-0.01	0.4

US accident: Even though the regression model results for the US accident data were statistically significant, the regression coefficient effect sizes are rather small. The magnitude or strength of the relationship between the features and the outcome variable is not strong, suggesting, the predictor variables have only a small impact on the outcome variable. However, in reality, weather conditions can be a significant factor in driving safety and accident prevention. Adverse weather conditions such as rain, snow, fog, heavy winds, and ice can impair visibility, reduce traction, and make it more difficult to control a vehicle [8]. This can increase the risk of accidents, particularly if drivers are not adequately prepared or trained to handle these conditions. In fact, according to the National Highway Traffic Safety Administration (NHTSA) in the United States, about

22% of the nearly 5.8 million vehicle crashes that occur each year are weather-related, resulting in approximately 6,000 fatalities and 445,000 injuries annually. Thus, while the impact of weather conditions may be small in a regression model, it can have significant real-world consequences for driving safety.

The p-values may not be reliable, and the coefficients and resulting predictions may also be questionable. Therefore, it is recommended to refine the model specifications and provide sound justifications for the predictions. A better understanding of the underlying data and potential sources of bias is necessary to improve the accuracy and reliability of the model.

In order to gain deeper insights into the relationship between the features and the target variable [11], we employed additional statistical analyses and specifically examined the bivariate relations within the dataset. In seatpos dataset, Age does not have a close association with the rest of the variables. Hipcenter seems to have a negative correlation with most variables, except Age. The remaining variables have strong positive associations with correlation values ranging $0.9 - 0.99$. Overall, the regression results show multicollinearity concerns. The overall results illustrate how a model's performance reduces and the explanation changes with highly correlated data. In the above-discussed example, the regression model could no longer provide correct predictions about the problem.

In the US accidents dataset that there is no significant correlation between the features examined. The weak association between these features and the target variable suggests that they may not be useful in predicting the values of the target. In light of these findings, it is clear that the impact of the features on the target variable is also weak.

Next, the variance inflation factor (VIF) values were estimated for the dataset with multicollinearity issue to identify which features are affected by multicollinearity and the strength of the correlation. Figure 1 shows the VIF for each feature. VIFs start at 1 and have no upper limit. A value of 1 indicates that there is no correlation between this independent variable and any others.

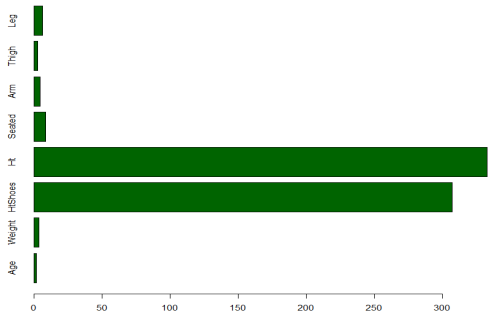


Fig. 1. Variance inflation factor (VIF) values

The VIF values for Age, Weight, Arm and Thigh are less than 5. The minimum VIF value is equal to 1.99 for Age. VIFs between 1 and 5 indicate that there is a moderate correlation. VIFs greater than 5 represent critical levels of multicollinearity where the coefficients are poorly estimated and the p-values are questionable. Features Ht and HtShoes have very large VIF values, 333 and 307, respectively. These indicators warrant corrective measures are necessary.

Given these results, it is important to consider whether to follow the model's suggestion that none of the features are relevant to the design of the car seat or predict the severity of the accident. In addition, there are other actions we can take to further assess the situation, such as removing some of the correlated features, applying LASSO and Ridge regression models, or linearly combining the features. For our models, none of these options are likely to enhance the explainability of our models. Alternatively, we could use the models with only a few significant predictors, but this may lead to a loss of explanatory power for our predictive problem, as the statistical insignificance of certain features may hinder our understanding of the model's predictions.

Ultimately, we should carefully consider our goals and priorities when deciding how to proceed with the model. It may be necessary to weigh the trade-offs between model complexity and interpretability, as well as the potential impact on the accuracy and robustness of our predictions.

4.3 Legal issues

In the previous subsection, we observed that the driver dimensions are insignificant when designing a car seat as well as we saw that the weather conditions have a very small impact on the accident severity. This outcome contradicts commonly held notions and studies in which significant relationships were observed between the driver's anthropometric parameters and the car seat design [13] and the weather conditions and the road accidents [12]. In this section, we delve into the legal issues related to XAI and the lack of explainability that can lead to different forms of decisions. The legal aspect of XAI gained attention when the lack of explainability resulted in ethical or legal issues [1]. Researchers like Gorski and Ramakrishna [3] have compared various XAI methods when applied to legal classification neural networks and provided a lawyer's perspective on the classification. Similarly, Waltl and Vogl [22] have considered different dimensions to capture transparency for legal informatics. However, it is important to note that legal and judicial reasoning in XAI varies from case to case, as different scenarios can result in different outcomes. The ultimate goal of legal XAI is to improve AI efficiency and avoid costly mistakes. In light of this, based on the case studies we have identified several legal issues that may arise due to incomplete or erroneous explainability.

1. Harm to physical integrity and security problems. This refers to the machine learning model's wrong decisions causing physical harm. In most countries, regulatory bodies are responsible for setting and enforcing safety standards for vehicles on the road. These standards can include regulations related to a

wide range of safety features, from airbags and seat belts to electronic stability control and brake systems. They can also include regulations specific to adverse weather conditions, such as the use of snow tires or the requirement to carry chains in certain areas. In our model, if a car seat is misdesigned, that poses an immediate risk to the driver, passenger, other drivers on the road, and pedestrians. Or for the next model, if a car manufacturer is found to have used a machine learning model that did not accurately account for the effects of snow or ice on the road, they may be fined for violating safety regulations related to winter driving. Inaccurate decisions could lead to various legal punishments such as product liability lawsuits, fines, and penalties depending on the jurisdiction and specific circumstances. In extreme cases, such as if a machine learning model's flawed or inaccurate decisions result in fatalities or other serious injuries, criminal charges such as manslaughter or negligent homicide may be brought against the car manufacturer or any other relevant parties involved in the design and production of the car and its components.

2. Unintentional misuse of data, which is the use of information in ways it wasn't intended to gain an undue advantage. A model is developed by one or more ML engineers, who intentionally or non-intentionally may develop a flawed model. It should be noted that the flawed model does not generate flawed decisions intentionally, but it works the way it was designed. For example, the seatpos data is intended to help design a safe and comfortable vehicle. However, the model using this dataset shows that the driver characteristics are not significant for the car seat design. Even when the data is misused unintentionally this could lead to social, financial, or personal damages. It is also possible that someone deliberately manipulates the data or the model to influence public opinion, advance a particular agenda or achieve a particular outcome, such as to cover up or downplay the severity of a weather-related incident. For example, if a transportation company intentionally underreports the severity of weather conditions to avoid delays or costs, they may be putting passengers at risk.
3. Lack of accountability and responsibility of machine learning models and to what degree the algorithm decisions are responsible for some problems. Assume the car seat is designed without considering the driver's characteristics and following what the original regression model predicts. In case the vehicle has issues, and the driver has some injuries, the critical question is who should be responsible for these injuries. From the legal point of view, the vehicle designer and manufacturer will be accountable for the damages. The case, when determining legal accountability for a machine learning model mispredicting a car accident due to weather conditions is a complex issue that requires a careful examination of the facts and circumstances of the case. It may involve multiple parties sharing varying degrees of responsibility, depending on the specific details of the case. The responsibility could be shared among the developers and designers of the machine learning model, who may be liable if they failed to properly test or validate the model, or if they did not account for known or foreseeable factors that could affect its

accuracy. The owners or operators of the vehicle could be held liable if they are aware of a known issue or defect with the machine learning model, and they continue to use it despite this knowledge.
Accurate and explainable machine learning could be very useful in preventing legal liabilities arising from accidents caused by faulty vehicle designs. By providing a clear understanding of the data and decisions that led to a design, machine learning can help ensure that vehicles are designed and built with safety in mind, thus reducing the risk of accidents and related legal issues.

5 Explainability analysis using Shapley net effects

Generating their own explanations is an indispensable requirement for intrinsically explainable models. In the above-analyzed cases, the data was highly relevant in predicting either the vehicle design or the likelihood of an accident. However, as we have observed, the regression model with highly correlated variables failed to provide correct explanations. Additionally, the model that did not include correlated variables failed to provide a clear explanation of how the features affected the prediction of the target variable. To improve the explainability of the regression models, we can use Shapley value net effects developed by Lipovetsky and Conklin [9]. Shapley value net effects determine the feature importance of the regression model with multicollinearity issue. The proposed approach employs a cooperative game theory solution concept called, Shapely value [19] (Eq. 2).

$$\phi_i(v) = \sum_{s \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!)}{|N|!} (v(S \cup \{i\}) - v(s)) \tag{2}$$

To determine the importance of a feature, the model's performance is evaluated by comparing the multiple determination R^2 value with and without that feature using the Shapley value (Eq. 3).

$$U_i = R^2 - R^2_{-i} \tag{3}$$

Equation 2 presents the Shapley value, which considers a coalition S that does not include i. The equation's initial component randomly selects a set size of $|S|$ out of $\{0, 1, 2, \ldots, |N|-1\}$, each having probability $\frac{1}{|N|}$ to be drawn. Afterwards, a subset of $N\{i\}$ of size $|S|$ is chosen. Marginal contribution of coalition member i is computed thereafter $v(S \cup \{i\})) - v(S)$. To learn more about cooperative game theory and Shapley value calculation, refer to [5]. Algorithm 1 provides a summary of the steps involved in computing the regression Shapley value net effects.

The algorithm begins by identifying various feature combinations and conducting regression models for each combination of features. The process concludes by presenting the model summaries and obtaining R^2 values. These new R^2 values are then employed as fresh data to specify the game in characteristic

Algorithm 1: Shapley value feature importance explanations

Data: $(X_1, X_2, ..., X_n)$
$r\_{}_{squared} \leftarrow []$;
combinations $C \leftarrow []$;
for X_i *in range* $(1 : n)$ **do**
  identify feature combinations;
  for *a combination* C *in the list of combinations* **do**
    estimate regression models;
    for this models extract $r_{squared}$ values
  end
end
Data: $(v\{1\}, v\{2\}, ..., v\{allfeatures\})$
define the game (n, V) in characteristic form;
compute Shapley regression values;
end When all possible combinations of coalitions are assessed;
Return feature importance values

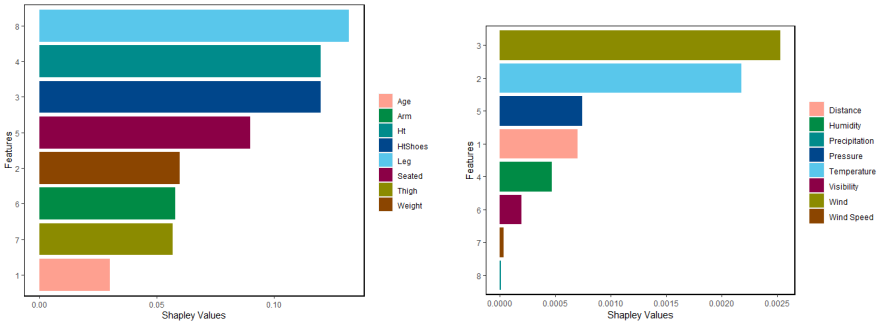
form. In the next phase of the algorithm, the game is defined for n features using the corresponding characteristic values or coalition values v in lexicographic order. The model that does not contain any features will have the characteristic value $v(0) = 0$, as the regression model with intercept only does not generate any R^2 result. The algorithm terminates when all permutations are assessed and yields the regression Shapley values, which represent the features' incremental contributions. The code and the datasets can be accessed online from https://github.com/grigoryangayane/RegressionShapley.

5.1 Shapley net effect results

This subsection presents the results of the regression Shapley value analysis for the seatpos and the US accident dataset following Algorithm 1. The model that has the lowest multiple determination value is the single variable model. By computing the Shapley values, we can determine the marginal contributions of all features to predict the target variables and understand their relative importance in designing the prediction model for both datasets.

In Figure 2-a, it can be observed that HtShoes, Ht, and Leg are the most significant variables in predicting the driver's car seat position. The respective marginal contributions of Age, Weight, HtShoes, Ht, Seated, Arm, Thigh, and Leg are $\phi_i = (0.032, 0.0661, 0.122, 0.124, 0.093, 0.058, 0.0572, 0.132)$. It is surprising to note that the arm and thigh length are among the least relevant features in explaining the predicted variable. Additionally, Age does not seem to be relevant in predicting the car seat position.

The US accident dataset analysis (Figure 2-b) suggests that wind chill is a significant predictor of accident severity. The temperature also plays a notable role in predicting accident severity, although to a lesser extent than wind



(a) Seatpos (b) US accident

Fig. 2. Shapley feature importance values

chill. Precipitation appears to be the least important feature when it comes to predicting the severity of accidents.

We can compare these results with outcomes obtained through alternative feature selection or explainable AI techniques. One widely popular approach is the use of SHAP (SHapley Additive exPlanations), which offers localized explanations and can be employed for comparative analysis. However, since our analysis focuses on global explanations, comparing it with local explainable results may not yield meaningful insights. Therefore, we employ a stepwise regression technique to identify the features selected using this method.

Stepwise regression is an iterative process used to select predictors for a predictive model. It involves adding and removing variables to identify the subset that results in the best-performing model. There are three strategies: forward selection, backward elimination, and a hybrid approach. The forward selection starts with no predictors and gradually adds the most influential predictors until no significant improvement is observed. Backward elimination begins with all predictors and removes the least influential ones until all remaining predictors are considered substantial. The hybrid approach combines both strategies.

In setpos dataset all three stepwise selection algorithms indicate that Age is a significant predictor, explaining a substantial amount of variance in the model. This aligns with the lack of a significant correlation between Age and other predictor variables. Additionally, Ht, HtShoes, and Leg are suggested as important features, consistent with the findings from Shapley Value analysis. Stepwise regression on the adult dataset reduced the number of unnecessary features. However, due to the presence of many categorical features in the dataset, stepwise regression kept the categories that were considered statistically significant for the model, resulting in the retention of numerous features as statistically significant. The retention of many features as statistically significant can make it challenging to discern the specific connections between individual features and the dependent variable. With categorical features, stepwise regression determines the significance of each category within a feature, rather than the feature as a

whole. Consequently, it may not provide a clear understanding of how specific categories or combinations of categories influence the outcome.

In a nutshell to prevent model mispredictions and mitigate legal risks, several steps can be taken, including implementing robust model development practices, maintaining transparent documentation, continuously monitoring and evaluating the model's performance, adhering to regulatory compliance, incorporating explainability components into the model, conducting risk assessments and mitigation, and seeking legal consultation. The inclusion of explainability components helps ensure that the decision-making process is understandable and transparent. Consulting with legal experts is crucial to ensure compliance with legal requirements and establish appropriate measures for preventing and addressing legal consequences resulting from incorrect regression model predictions.

6 Conclusion

The conclusion of our paper highlights the importance of explainable artificial intelligence (XAI) in promoting trust, understanding, and reliability in AI-powered decisions. As we have observed in the last several years, there has been a growing interest in advanced AI systems achieving impressive task performance. However, there has also been an increased awareness of their complexity and challenging consequences of their possibly limited understandability to humans. Our paper addresses the need for explainability, especially for inherently explainable models, and discusses legal issues associated with model misprediction. We have demonstrated through a case study of multiple linear regression models in the transportation domain, both with and without multicollinear features, that without explainability, the models fail to provide accurate predictions and relationships between each feature and the target.

Our findings are in line with the growing research directions toward Responsible AI (RAI). The concept of XAI provides a foundation for transparency and understandability, which is essential for value alignment and human centricity. However, we believe that responsibility and accountability should primarily be considered from the legal perspective, as the operation of AI-based systems poses actual challenges to rights and freedoms of individuals.

The legal challenges related to AI decision-making require careful consideration to ensure that AI is used in a way that is safe, fair, and transparent. The work aims to integrate methodological AI, as well as ethical and legal perspectives, to address questions such as the legal consequences of black-box AI systems, criteria of legal responsibility, and possible applications of XAI systems in the area of legal policy deliberation and legal practice. By doing so, we hope to contribute to the development of explainable, transparent, and responsible AI systems that can be effectively utilized in different spheres of societal life.

References

1. Bibal, A., Lognoul, M., De Streel, A., Frénay, B.: Legal requirements on explainability in machine learning. Artificial Intelligence and Law **29**(2), 149–169 (2021)

2. Faraway, J.: Linear models with r. crc press. Boca Raton, Florida (2014)
3. Górski, Ł., Ramakrishna, S.: Explainable artificial intelligence, lawyer's perspective. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. pp. 60–68 (2021)
4. Grigoryan, G.: Explainable artificial intelligence: Requirements for explainability. In: Proceedings of the 2022 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. pp. 27–28 (2022)
5. Grigoryan, G., Collins, A.J.: Game theory for systems engineering: a survey. International Journal of System of Systems Engineering **11**(2), 121–158 (2021)
6. Gunning, D., Aha, D.: Darpa's explainable artificial intelligence (xai) program. AI magazine **40**(2), 44–58 (2019)
7. Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H.: The elements of statistical learning: data mining, inference, and prediction, vol. 2. Springer (2009)
8. Khattak, A.J., Kantor, P., Council, F.M.: Role of adverse weather in key crash types on limited-access: roadways implications for advanced weather systems. Transportation research record **1621**(1), 10–19 (1998)
9. Lipovetsky, S., Conklin, M.: Analysis of regression in game theory approach. Applied Stochastic Models in Business and Industry **17**(4), 319–330 (2001)
10. Lipton, Z.C.: The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. Queue **16**(3), 31–57 (2018)
11. Lynch, C.J., Gore, R., Collins, A.J., Cotter, T.S., Grigoryan, G., Leathrum, J.F.: Increased need for data analytics education in support of verification and validation. In: 2021 Winter Simulation Conference (WSC). pp. 1–12. IEEE (2021)
12. Malin, F., Norros, I., Innamaa, S.: Accident risk of road and weather conditions on different road types. Accident Analysis & Prevention **122**, 181–188 (2019)
13. Mohamad, D., Deros, B.M., Daruis, D.D., Ramli, N.F., Sukadarin, E.H.: Comfortable driver's car seat dimensions based on malaysian anthropometrics data (2016)
14. O'Sullivan, S., Nevejans, N., Allen, C., Blyth, A., Leonard, S., Pagallo, U., Holzinger, K., Holzinger, A., Sajid, M.I., Ashrafian, H.: Legal, regulatory, and ethical frameworks for development of standards in artificial intelligence (ai) and autonomous robotic surgery. The international journal of medical robotics and computer assisted surgery **15**(1), e1968 (2019)
15. Robaldo, L., Pacenza, F., Zangari, J., Calegari, R., Calimeri, F., Siragusa, G.: Efficient compliance checking of rdf data. Journal of Logic and Computation (to appear) (2023)
16. Robaldo, L., Villata, S., Wyner, A., Grabmair, M.: Introduction for artificial intelligence and law: special issue "natural language processing for legal texts". Artificial Intelligence and Law **27**(2) (2019)
17. Rodrigues, R.: Legal and human rights issues of ai: Gaps, challenges and vulnerabilities. Journal of Responsible Technology **4**, 100005 (2020)
18. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**(5), 206–215 (2019)
19. Shapley, L.: A value for n-person games. Contributions to the Theory of Games **28**(2), 307–317 (1953)
20. Shelden, C.H.: Prevention, the only cure for head injuries resulting from automobile accidents. Journal of the American Medical Association **159**(10), 981–986 (1955)
21. Shin, S.Y., Lee, S.: Correlation between car accident and car color for intelligent service. Journal of Intelligence and Information Systems **19**(4), 11–20 (2013)
22. Waltl, B., Vogl, R.: Explainable artificial intelligence the new frontier in legal informatics. Jusletter IT **4**, 1–10 (2018)

LexGPT 0.1: pre-trained GPT-J models with Pile of Law

Jieh-Sheng Lee[0000−0002−0990−6170]

National Yang Ming Chiao Tung University School of Law
No. 1001, Daxue Rd. East Dist., Hsinchu City 300093, Taiwan
jasonlee@nycu.edu.tw

Abstract. This research aims to build generative language models specialized for the legal domain. The manuscript presents the development of LexGPT models based on GPT-J models and pre-trained with Pile of Law. The foundation model built in this manuscript is the initial step for the development of future applications in the legal domain, such as further training with reinforcement learning from human feedback. Another objective of this manuscript is to assist legal professionals in utilizing language models through the "No Code" approach. By fine-tuning models with specialized data and without modifying any source code, legal professionals can create custom language models for downstream tasks with minimum effort and technical knowledge. The downstream task in this manuscript is to turn a LexGPT model into a classifier, although the performance is notably lower than the state-of-the-art result. How to enhance downstream task performance without modifying the model or its source code is a research topic for future exploration.

Keywords: Natural Language Processing · Natural Language Generation · Generative Model · Legal Text.

1 Introduction

The codename "LexGPT" in this research refers to the development of GPT (Generative Pre-trained Transformer) [12] models specialized for the legal domain. The objective is to create models that can assist legal professionals in performing various legal tasks in the future. In this manuscript, "LexGPT 0.1" refers to the models based on GPT-J [19] and pre-trained with Pile of Law [6] dataset. LexGPT as a foundation model is essential for the development and success of future applications in the legal domain, including those might build on InstructGPT [14] or ChatGPT [13] models. The progress made in this manuscript represents the initial step towards developing future models and applications. The pre-trained LexGPT models will be released for further research and development.

To facilitate the adoption of the LexGPT models by legal professionals, a downstream classification task is presented as an example. This task involves fine-tuning the models without the need to add a classification layer, thereby

eliminating the need for enhancing the source code of the model. In the legal domain, developing language models or applications may require technical skills that legal professionals may not possess. This can create a significant entry barrier for those who want to leverage LexGPT models in this research. To overcome this barrier, an objective of this manuscript is to explore the possibilities of leveraging language models under the "No Code" idea. In computer science, "No Code" refers to a way of building software applications without requiring extensive knowledge or experience in programming languages. The idea is intended to democratize access to technology by providing users with tools, templates, and interfaces to create applications easily and quickly. In this manuscript, the downstream tasks are conditioned on fine-tuning models with specialized data and without modifying any source code. By doing so, legal professionals can create custom language models based on pre-trained LexGPT models with minimum effort and technical knowledge.

2 Related Work

Language models have proven to be effective in many domains and are beginning to make an appearance in the field of law. For example, LEGAL-BERT [2] shows that pre-training BERT from scratch with legal corpuses performs better on average. Without domain adaptation, the authors found that the previous pretraining and fine-tuning do not always generalize well in the legal domain. In [5], the authors fine-tuned a popular BERT language model trained on German data (German BERT) for Named Entity Recognition (NER) tasks in the legal domain. In [6], a BERT-large equivalent model was trained to predict whether a paragraph should use pseudonymity. Legal datasets are scarce and expensive because of the complexity and specialty. The authors of [6] curated a $\sim$256GB dataset of legal and administrative text, which is called Pile of Law. The dataset is intended for learning responsible data filtering from the law. In [3], the authors introduced the Legal General Language Understanding Evaluation (LexGLUE) benchmark, a collection of datasets for evaluating model performance across a diverse set of legal NLU tasks. The models evaluated in [3] are BERT-based models. For processing long legal documents, the authors in [10] modified a Longformer warm-started from LegalBERT and modified LegalBERT to use TF-IDF representations.

Most language models utilized in the legal field are built upon BERT model. While GPT models are adept at generative tasks, they are not commonly examined as a foundation model for legal tasks in academics. In [9], the author fine-tuned OpenAI GPT-2 [12] models for patent claim generation. However, the models are specific to the patent field only. In [11], the author built LawGPT 1.0 as a virtual legal assistant built by fine-tuning GPT-3 for the legal domain. The author provided a brief overview but the detailed information about the model is protected by a non-disclosure agreement (NDA) and cannot be disclosed. In [6], the authors noted that the Pile of Law dataset can be used in the future for pretraining legal-domain language models. Given the limited explo-

ration of GPT models in the legal domain, this research is motivated to undertake pre-training of GPT models on the Pile of Law dataset for downstream legal tasks. The pre-training can serve as a precursor to the subsequent development of advanaced models or applications. For evaluating the performance of natural language processing (NLP) models, the General Language Understanding Evaluation (GLUE) [17] benchmark is a popular benchmark. In the legal domain, LexGLUE [3] is a benchmark dataset for evaluating legal language understanding. Language models in [3] rely on BERT only and no GPT models are included. For pre-training sizeable GPT models with open-sourced code, the repositories available in public include GPT-J-6B [19] (using TPUs), GPT-NeoX-20B [1] (using GPUs), and Open Pre-trained Transformers (OPT) [21] (using GPUs).

3 Implementation

3.1 Objectives

The primary objective of this manuscript is to build LexGPT as foundation models by pre-training GPT models exclusively with legal text. It is important for legal professionals that the model generates only legal text. These pre-trained models will serve as the basis for downstream tasks. The second objective is to evaluate the performance of downstream tasks by fine-tuning the LexGPT models and using the LexGLUE benchmark. The tasks are developed under the purposeful constraint that no additional source code or new layers are added to the model. Lastly, this manuscript aims to document instances of failure and lessons learned so that subsequent researchers may discover improved solutions.

3.2 Pre-trained models

Why GPT-J-6B? At the beginning of this research, the GPT-J-6B model in [19] was released as the largest pre-trained model available to the public. GPT-J-6B is a transformer model trained using Mesh Transformer JAX and developed by EleutherAI, an independent research organization focused on advancing open-source artificial intelligence. The model implements the GPT architecture developed by OpenAI. According to [19], GPT-J-6B has achieved impressive results on various language tasks, such as text generation, translation, and question answering. The size of the GPT-J-6B model is also suitable for quicker iterations and proof of concept. The model runs on TPU (Tensor Processing Unit) instead of GPU.

Why pre-training? The primary objective of this manuscript is to construct GPT models using the Pile of Law dataset. The codebase in [19] provides a guide for fine-tuning [18] the GPT-J-6B model. However, since the original GPT-J-6B model's training data does not solely include legal text, a fine-tuned model could generate non-legal text, which would be of little use to legal professionals. Hence,

after referencing the fine-tuning guide, the models in this manuscript are pre-trained from scratch instead. These pre-trained models can serve as the foundation models for downstream tasks and future applications in the legal domain. For instance, an application such as ChatGPT requires a foundation model for training with reinforcement learning from human feedback (RLHF) [4]. The concept of incorporating human feedback to solve deep reinforcement learning tasks was introduced by OpenAI. The idea paved the way for developing InstructGPT, and now ChatGPT. To implement RLHF with legal professionals' feedback, the pre-trained LexGPT models serve as the first step towards this goal.

Dataset In this manuscript, the Pile of Law dataset from [6] is utilized to pre-train GPT-J models. The dataset was initially released with a size of 256G and is still expanding. As of the time of writing, the estimated size of the dataset is 291.5 GiB. The first release of the dataset is employed in this study. The Dataset Card [7] indicates that the dataset can be used for pre-training language models as a key direction in access-to-justice initiatives.

Tokenizer The GPT-J-6B model is trained with a tokenization vocabulary of 50257, using the same BPE (Byte Pair Encoding) as GPT-2 and GPT-3. To enhance the accuracy of language models, the LexGPT models discussed in this manuscript utilize domain-specific vocabularies trained from the Pile of Law. One of the tokenizers is trained with a vocabulary size of 50257, while the other has a reduced size of 25129, representing half of the former. These two tokenizers are provided for experiments.

Model sizes & Training Losses The training losses of LexGPT-6B models, using different tokenizers, are depicted in Fig.1. The model using the original tokenizer in GPT-J-6B is represented by the curve "6B". However, it was observed that the default learning rate (ranging from *1.2e-4* to *1.2e-5*) for GPT-J-6B was inadequate, as illustrated in Fig.1. To address this, a lower learning rate (ranging from *0.6e-4* to *0.6e-5*) was experimented, and the result is shown by the curve "6B_v2". Subsequent models utilized the same lower learning rate. Additionally, the curve "6B_bpe_25129" represents the LexGPT-6B model utilizing a domain-specific tokenizer with a vocabulary size of 25129. The curve "6B_bpe_50257" represents the model using a domain-specific tokenizer with a vocabulary size of 50257.

Using the domain-specific tokenizers and the same range of lower learning rates, the pre-training process extended to include models of size 1.6B and 456M. The corresponding training losses are displayed in Fig. 2. Notably, the final training losses did not exhibit significant gains, despite differences in model size. For all models, the training step is 350,000 and the maximum sequence length of the model is 2,048. These settings are the default values in the configuration file "6B_roto_256.json" provided in GPT-J-6B codebase. In this research, the Pile of Law contains approximately 60 billion tokens after tokenization. By utilizing

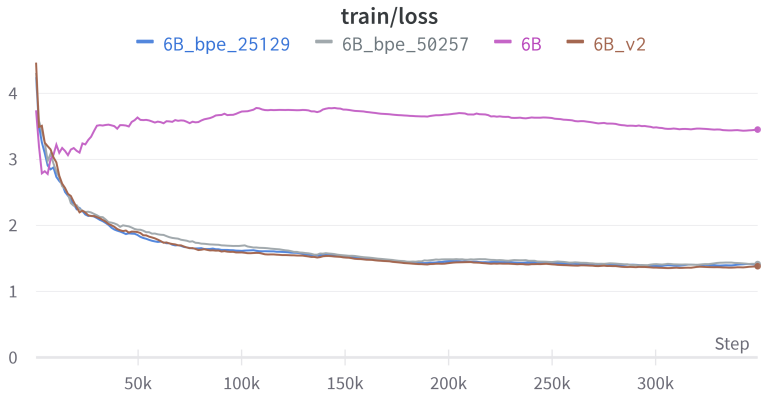


Fig. 1. Training loss

TPU v3-8 and setting the batch size to 8, the pre-training process covers about 10.6% of all tokens.

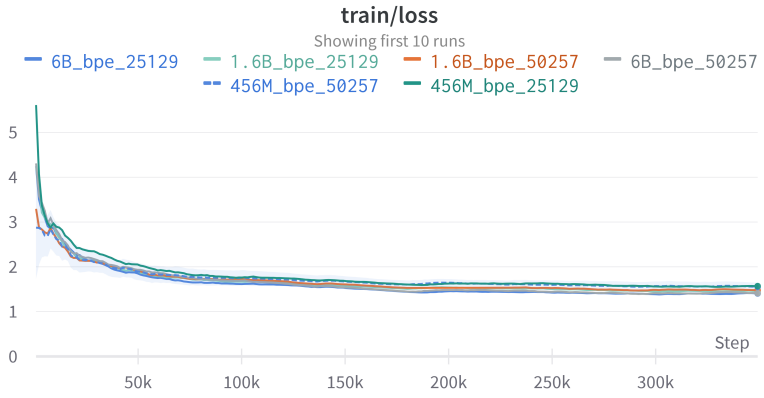


Fig. 2. Training loss

Release Language models specialized in the legal domain have the potential to enhance access to justice. The pre-trained LexGPT models, along with domain-specific tokenizers, tokenized training and validation datasets, configuration files, and relevant source code, will be publicly available upon publication of this manuscript. However, it should be noted that language models may make factual

mistakes and experience hallucinations. Therefore, legal professionals are recommended as the initial target users of legal applications based on these models, as their legal knowledge can help filter out any mistakes and hallucinations.

3.3 Downstream Tasks

In this manuscript, the downstream tasks are single-labe classification tasks and are achieved by fine-tuning LexGPT models using task-specific text and labels. Typically, for classification tasks, additional task-specific layers are added on top of an existing model, and the entire custom setup is fine-tuned end-to-end. This requires modifying the source code of the original model or adding new code to wrap around the extracted body of the model. Both involves coding if no classification function is provided in the source code of the original model for customization. However, one objective of this manuscript is to adhere to the "No Code" idea, and thus, the aforementioned coding approaches are not considered. Instead, to turn a LexGPT model into a classifier, the model is fine-tuned with training data in the format of "(text)< |label| >(label)", where "< |label| >" is a special tag that concatenates the text and label for training. At inference, the fine-tuned model is utlized to predict the correct label by prompting "(text)< |label| >" to the model. Being able to predict the correct label makes the fine-tuned model a classifier. In this way, one can create a classifier without modifying any source code or structure of the model.

Dataset The benchmark in this manuscript comes from the LexGLUE [3]. LexGLUE is based on seven existing legal NLP datasets, selected using criteria largely from SuperGLUE [16]. The tasks they address have been simplified to make it easier for generic models to address all tasks. In this research, the downstream task focuses on the LEDGAR (LEDGAR (Labeled EDGAR) dataset and the CaseHOLD (Case Holdings on Legal Decisions) dataset in LexGLUE. As stated in [3], the LEDGAR dataset is a dataset for contract provision (paragraph) classification. The contract provisions come from contracts obtained from the US Securities and Exchange Commission (SEC) filings. The original dataset includes approximately 850k contract provisions labeled with 12.5k categories. In LexGLUE, the authors use a subset of the original dataset with 80k contract provisions, considering only the 100 most frequent categories as a simplification. The auhtors split the new dataset chronologically into training (60k, 2016–2017), development (10k, 2018), and test (10k, 2019) sets. Each label represents the single main topic of the corresponding contract provision, i.e., it is a single-label multi-class classification task. The number of classes is 100. As for the CaseHOLD [22] dataset, it contains approximately 53k multiple choice questions about the holdings of US court cases from the Harvard Law Library case law corpus. The input consists of an excerpt from a court decision, containing a reference to a particular case, where the holding statement is masked out. The task is to identify the correct (masked) holding statement from a selection of five choices. The dataset is split in training (45k), development (3.9k), test (3.9k) sets.

In this study, the remaining five datasets in LexGLUE are set aside for future experiments due to the following reasons: Firstly, the text in the ECtHR (A and B) and SCOTUS datasets is significantly lengthier than the input sequence length of LexGPT models. In [3], the authors employ a hierarchical variant of each pre-trained Transformer-based model that has not been designed for longer text. Since no source code modification is one of the objectives in this manuscript, these three datasets are skipped. Secondly, the EUR-LEX and UNFAIR-ToS datasets are tasks of multi-label classification. Since a generative language model predicts the next token based on previouis tokens, predicting a label may attend inadequately to a previous label in the multi-label settings. Predicting a label should base on its input text only. How to formulate a multi-label classification task and fit the sequential nature of a generative language model is a subject for future research.

Experiment 1: LEDGAR In this experiment, the LexGPT models of size 456M and 1.6B (bpe_25129) are fine-tuned with the LEDGAR training data in [3] once. The 456M model obtained a *micro-F1* score of 83.5% and a *macro-F1* score of 72.4%. The 1.6B model obtained a *micro-F1* score of 83.9% and a *macro-F1* score of 74.0%. These numbers are not state-of-the-art results. In [3], the highest *micro-F1* score is 88.3% based on the CaseLaw-BERT model and the highest *macro-F1* score is 83.1% based on the DeBERTa model. It is noted that the original LEDGAR dataset described in [15] is for multi-label classification. In [3], the dataset is simplified as a single-label multi-class classification task. It remains to be investigated in the future whether LexGPT models would outperform the state-of-the-art on the original LEDGAR dataset in a multi-label setting.

Experiment 2: CaseHOLD In this experiment, the LexGPT models of size 456M and 1.6B (bpe_25129) are fine-tuned with the CaseHOLD training data once, resulting in accuracies of 49.6% and 27.6%, respectively. The CaseHOLD task is to identify the correct holding in a prompt from a selection of five choices. In this study, the multiple choice task is converted into a multiple binary classification task, and the accurarcy is calculated based on the top probability of choices among the multiple binary classification tasks. Although the accuracy (49.6%) of the 456M model is better than random guesses (20%), it is still significantly lower than the state-of-the-art result. In [3], the CaseHOLD task was performed using the CaseLaw-BERT model, which achieved the highest accuracy of 75.4%. According to the implementation in [3], each training instance consists of the prompt and one of the five candidate answers. The top-level representation h[cls] of each pair is fed to a linear layer to obtain a logit, and the five logits are then passed through a softmax yielding a probability distribution over the five candidate answers. Future research is required to determine whether specialized training data format can help narrow the performance gap of LexGPT models.

4 Conclusion and Future Work

The major contribution made in this study is the pre-trained LexGPT models using the Pile of Law dataset. The pre-trained LexGPT models will be released at [8]. Such foundation models can pave the development of InstructGPT-based or ChatGPT-based applications for the legal domain in the future. In addition, this study aims to provide legal professionals with a simple way to create custom language models without the need to modify its source code. The experimental results demonstrate that the pre-trained LexGPT models can be fine-tuned using task-specific data and labels to function as a classifier. However, despite the minimal effort required, the performance of the fine-tuned GPT model falls short compared to the conventional approach of modifying source code and adding a new classification layer to the model. It is noted that most classification tasks in the legal field are built upon BERT or similar models. How to utilize GPT models are less explored. It remains to be seen whether adding a new classification layer to LexGPT models can outpuform BERT-based models. Another area for future exploration is to investigate, under the "No Code" condition, whether the Chain-of-Thought (CoT) [20] ability of large language models can enhance the effectiveness of the classifiers in this study if training data is provided in CoT format.

Acknowledgements The research reported in this manuscript has been funded by the Ministry of Science and Technology (MOST) in Taiwan (Project ID: 111-2222-E-A49-005). In addition, the author would like to thank TensorFlow Research Cloud (TRC) greatly for providing powerful computational resources to make this research possible.

References

1. Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., ..., Weinbach, S.: GPT-NeoX-20B: An open-source autoregressive language model. In: Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models. pp. 95–136. Association for Computational Linguistics, virtual+Dublin (May 2022). https://doi.org/10.18653/v1/2022.bigscience-1.9, https://aclanthology.org/2022.bigscience-1.9
2. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: The muppets straight out of law school. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 2898–2904. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.findings-emnlp.261, https://aclanthology.org/2020.findings-emnlp.261
3. Chalkidis, I., Jana, A., Hartung, D., Bommarito, M., Androutsopoulos, I., Katz, D., Aletras, N.: LexGLUE: A benchmark dataset for legal language understanding in English. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume

1: Long Papers). pp. 4310–4330. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.acl-long.297, https://aclanthology.org/2022.acl-long.297
4. Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf
5. Darji, H., Mitrović , J., Granitzer, M.: German BERT model for legal named entity recognition. In: Proceedings of the 15th International Conference on Agents and Artificial Intelligence. SCITEPRESS - Science and Technology Publications (2023). https://doi.org/10.5220/0011749400003393, https://doi.org/10.5220/0011749400003393
6. Henderson*, P., Krass*, M., Zheng, L., Guha, N., Manning, C., Jurafsky, D., Ho, D.E.: Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset (2022)
7. HuggingFace: Dataset card for pile of law. https://huggingface.co/datasets/pile-of-law/pile-of-law (2022)
8. Lee, J.S.: LexGPT repository. https://github.com/jiehsheng/LexGPT (2023)
9. Lee, J.S., Hsiang, J.: Patent claim generation by fine-tuning openai gpt-2. World Patent Information **62**, 101983 (2020). https://doi.org/https://doi.org/10.1016/j.wpi.2020.101983, https://www.sciencedirect.com/science/article/pii/S0172219019300766
10. Mamakas, D., Tsotsi, P., Androutsopoulos, I., Chalkidis, I.: Processing long legal documents with pre-trained transformers: Modding LegalBERT and longformer. In: Proceedings of the Natural Legal Language Processing Workshop 2022. pp. 130–142. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid) (Dec 2022), https://aclanthology.org/2022.nllp-1.11
11. Nguyen, H.T.: A brief report on lawgpt 1.0: A virtual legal assistant based on gpt-3 (2023). https://doi.org/10.48550/ARXIV.2302.05729, https://arxiv.org/abs/2302.05729
12. OpenAI: Better Language Models and Their Implications. https://openai.com/blog/better-language-models (Feb 2019)
13. OpenAI: Introducing ChatGPT. https://openai.com/blog/chatgpt (Nov 2022)
14. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., ..., Lowe, R.: Training language models to follow instructions with human feedback (2022). https://doi.org/10.48550/ARXIV.2203.02155, https://arxiv.org/abs/2203.02155
15. Tuggener, D., von Däniken, P., Peetz, T., Cieliebak, M.: LEDGAR: A large-scale multi-label corpus for text classification of legal provisions in contracts. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. pp. 1235–1241. European Language Resources Association, Marseille, France (May 2020), https://aclanthology.org/2020.lrec-1.155
16. Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. Curran Associates Inc., Red Hook, NY, USA (2019)
17. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 353–355. Association for Computational Lin-

guistics, Brussels, Belgium (Nov 2018). https://doi.org/10.18653/v1/W18-5446, https://aclanthology.org/W18-5446
18. Wang, B., Biderman, S., de la Rosa, J.: How to Fine-Tune GPT-J. https://github.com/kingoflolz/mesh-transformer-jax/blob/master/howto_finetune.md (Dec 2021)
19. Wang, B., Komatsuzaki, A.: GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax (May 2021)
20. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ..., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models (2023)
21. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., ..., Zettlemoyer, L.: Opt: Open pre-trained transformer language models (2022). https://doi.org/10.48550/ARXIV.2205.01068, https://arxiv.org/abs/2205.01068
22. Zheng, L., Guha, N., Anderson, B.R., Henderson, P., Ho, D.E.: When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. p. 159–168. ICAIL '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3462757.3466088, https://doi.org/10.1145/3462757.3466088

Classifying norms via pre-trained language models: experiments on the DAPRECO-KB*

Davide Liga[1] and Livio Robaldo[2]

[1] Individual and Collective Reasoning Group (ICR), University of Luxembourg, 6 av. de la Fonte, 4364, Esch-sur-Alzette, Luxembourg. davide.liga@uni.lu
[2] Legal Innovation Lab Wales, Swansea University, Singleton Park, Sketty, Swansea SA2 8PP, UK. livio.robaldo@swansea.ac.uk

Abstract. In this paper, we investigate the use of pre-trained language models to classify legal rules, i.e., regulative rules (obligations and permissions) and constitutive rules. We train and test five pre-trained language models on the DAPRECO knowledge base [25], which encodes the norms in the GDPR in LegalDocML [23] and LegalRuleML [3], two widely used XML standards for the legal domain. We use the LegalDocML and LegalRuleML annotations provided in [25] to fine-tune the pre-trained language models. Our results show that all of them are capable to learn how to classify legal rules even on small amount of data. In addition, we show that the (two) pre-trained language models using GPT-3 [24], perhaps the most currently discussed language model in the field of AI after the release of OpenAI, significantly outperform the (three) pre-trained language models based on BERT [9]. This paper is indeed the first attempt to fine-tune the GPT-3 on the recognition of legal rules. Our results confirm GPT-3's superiority with respect to its predecessors, namely the language models based on BERT.

Keywords: Legal rule classification· Transformers· Deep learning

1 Introduction

The automated classification of legal rules from existing legislation is an important research direction for the whole AI&Law community, in that it will enable the development of advanced legal document management systems (cf. [4]). Since legislation is originally available in natural language, Natural Language Processing (NLP) is one of the main AI technologies used in AI&Law [20],[26].

* Davide Liga was supported by the project INDIGO, which is financially supported by the NORFACE Joint Research Programme on Democratic Governance in a Turbulent Age and co-funded by AEI, AKA, DFG and FNR and the European Commission through Horizon 2020 under grant agreement No 822166. Livio Robaldo has been supported by the Legal Innovation Lab Wales operation within Swansea University's Hillary Rodham Clinton School of Law. The operation has been part-funded by the European Regional Development Fund through the Welsh Government.

Recent developments in NLP have shown the groundbreaking power of deep learning language model based on the self-attention mechanism, known as "Transformers" [31]. Transformers were created in 2017 in the research department of Google LLC and soon became the reference model for NLP. One year later, the research led to the development of *pre-trained* systems such as Google BERT (Bidirectional Encoder Representations from Transformers) [9] and OpenAI's GPT (Generative Pre-trained Transformer) [24].

Pre-trained language model have shown the ability to be applied to a wide range of NLP tasks, overcoming the state of the art in many NLP challenges. On the other hand, the use of pre-trained language models in AI&Law will foster a deeper integration of "bottom-up" data-driven AI with "top-down" symbolic AI, i.e., an integration of PLMs with the most recent theoretical results in formal deontic logic and argumentation [33, 2, 30].

In this paper, we focus on classification of legal rules, i.e., regulative and constitutive rules, that we may find in existing legislation. The task has received little attention by the scientific community in NLP for Artificial Intelligence and Law, despite its crucial role for LegalTech applications, as explained earlier.

Specifically, this paper presents an experiment on the DAPRECO knowledge base (DAPRECO-KB, for short) [25], a repository formalizing the norms of the General Data Protection Regulation (GDPR) while classifying them into obligations, permissions, and constitutive rules. Formalizations are encoded in LegalRuleML [3] and connected to the representation of the GDPR in LegalDocML [23]. LegalRuleML and LegalDocML are two widely used XML standards for the legal domain. Further details about the DAPRECO-KB are shown below in section 4; in our experiments, we use the XML annotations in the DAPRECO-KB to fine-tune our considered pre-trained language models.

In particular, we fine-tuned on the DAPRECO-KB the three pre-trained language models developed in [18], which are based on BERT and other two similar models (DistilBERT [28], and LegalBERT [8]), we developed and fine-tune on the DAPRECO-KB two new versions of GPT-3, and, finally, we compare the results. As anticipated in the abstract, the two fine-tuned versions of GPT-3 significantly outperform the three ones based on BERT.

The rest of the paper is organized as follows. In the Section 2, we will describe some related works, while Section 3 will briefly give an overview of our method, including a short introduction on both the data extraction technique (subsection 3.1) and the classification technique (subsection 3.2). In the following two sections, we will give a more exhaustive description of the retrieved data (Section 4), and a detailed report on the experimental settings with their respective results (Section 5). In particular, we present four experimental scenarios and two prompt strategies, i.e., eight settings in total. Section 6 concludes the paper.

2 Related Works

There have been (relatively few) attempts in literature to employ NLP methodologies to automatically detect rules in legal documents. Among these first at-

tempts, one can find studies tackling the classification of deontic elements [10] as parts of a wider range of targets [15],[11], [19]. Among these first attempts to classify obligations from legal texts there is [15]. These old methods employed word lists, grammars and heuristics to extract obligations among other targets such as rights and constraints.

Another work which tackled the classification of deontic statements is [32], which focused on the German tenancy law and classified 22 classes of statements (among which there were also prohibitions and permissions). The method used active learning with multinomial naive bayes, logistic regression and multi-layer perceptron classifiers, on a corpus of 504 sentences.

Similar approaches are [11], who used Machine Learning (ML) to extract six classes of normative relationships (prohibitions, authorizations, sanctions, commitments, and powers) and [21], who classify legal sentences in financial legislation using a Bi-LSTM architecture, with a training dataset containing 1,297 instances (596 obligations, 94 prohibitions, and 607 permissions).

The results shown in the mentioned ML approaches were adequate but not fully satisfactory. What prevented these methods from achieving better results was the lack of more available data for training the models, as well as data designed *ad hoc* for the classification of legal rules.

It is in fact well-known that approaches based on machine learning ought to be trained on large datasets, in order to make good predications [5], especially those employing deep neural architectures, which notoriously need huge amounts of data [21], [7]. Nevertheless, it is likewise well-known that in the legal domain large annotated datasets are usually unavailable. Indeed, it may be argued that even the original amount of legislative documents to be indeed too small for being considered as "Big Data" [1].

Creating such datasets is not only time-consuming but also costly because it requires domain expertise, which, on the one hand, is not always available and, on the other hand, in turn requires specific training of the domain experts, who are usually unfamiliar with annotation formats and technical details.

In this regard, pre-trained language models appear to be an optimal solution with respect to the trade-off between accuracy and unavailability of large training datasets. The main feature of pre-trained language models is known as "transfer learning", which indicates the idea that we can use these models by transferring what they "learnt" during their pre-training phase to downstream tasks and downstream data. Thanks to transfer learning, these models are able to achieve impressive results *even on small datasets*.

We believe that transfer learning is the main reason behind the increasing popularity of pre-trained language models and we advocate their use in the legal domain in particular, due to the aforementioned lack of annotated datasets.

Examples of past approaches employing pre-trained language models in the legal domain are [29], [14], and [18]. [29] used four pre-trained architectures (BERT, DistilBERT, RoBERTa, and ALBERT) but focused just on the binary detection duties vs non-duties. [14] also focused on permissions, achieving an average precision and recall of 90% and 89.66% respectively. Finally, [18] showed

how to use BERT, DistilBERT, and LegalBERT to classify legal rules, inspired in turn by previous positive results using Tree Kernel algorithms [17]

However, no one attempted to use generative pre-trained language models, such as GPT-3, the model at the basis of ChatGPT [6]. This work will cover this gap: we will both fine-tune the three BERT-based pre-trained language models in [18] and other two ones based on GPT-3 on the LegalRuleML annotations within the DAPRECO-KB as well as the references to the LegalDocML annotations of the original textual norms from the GDPR.

3 Methodology

As stated earlier, we want to use XML legal standards, namely LegalRuleML and LegalDocML, in combination with GPT-3, currently the most powerful and discussed pre-trained language model. Specifically, this paper will show the potential of using legal XML documents as source of data for applying GPT-3 on downstream tasks such as legal rule classification.

This task consists in classifying single legal sentences or single legal provisions from the DAPRECO-KB. These contain deontic statements such as obligations, prohibitions and permissions, constitutive rules, and legal provisions which do not contain any kind of rule, which we will call as "non-rules". Our experiments present in particular four different scenarios of classification:

(1) Rule vs Non-rule
(2) Deontic vs Non-deontic
(3) Obligation vs Permission vs Non-deontic
(4) Obligation vs Permission vs Constitutive Rule vs Non-rule

Our objective is twofold. On the one hand, we aim at showing that LegalRuleML and LegalDocML can be combined to feed generative AI machine learning algorithms with reliable data for the classification of legal rules. On the other hand, it aims at testing the use of transfer learning on the task of rule classification.

The first objective (i.e., combining LegalRuleML and LegalDocML) is related to the methodology that has been used to extract the legal knowledge and data. The second objecting (i.e., the use of transfer learning as machine learning algorithm) is related to the methodology for the classification.

The combination of these two methodological objectives led to the definition of our Hybrid AI approach, since it combines symbolic knowledge with subsymbolic knowledge (cf. [12], [27], and [16]).

3.1 Data extraction method

The idea underneath the methodology for the first objective is that combining LegalRuleML and LegalDocML is a powerful and convenient solution to extract labelled data for the classification of rules and deontic modalities.

LegalRuleML describes the logical content of the norms, i.e., it contains machine-readable representations of the legal rules. In addition, it links these

logical representations to the original natural language text in the regulations, namely the ID of the LegalDocML paragraph or point enclosing this text.

LegalDocML contains crucial pieces of information not only about the legal document, but also about the structure of natural language denoting the logical content. Thus, LegalDocML facilitates the reconstruction of the natural language contents, especially in those cases where the logical information is split across different structural portions of the legal source.

In this work, these two formats have been used to create a dataset, where natural language sentences are taken (and sometimes reconstructed) from LegalDocML, while the classes are extracted from LegalRuleML. As said above, the dataset was created from the DAPRECO-KB, the biggest existing LegalRuleML repository, which logically represents the norms of the GDPR and contains the references to the IDs of the LegalDocML annotations of the regulation. Figure 1 shows the methodology to create the datasets out of the two XML standards.

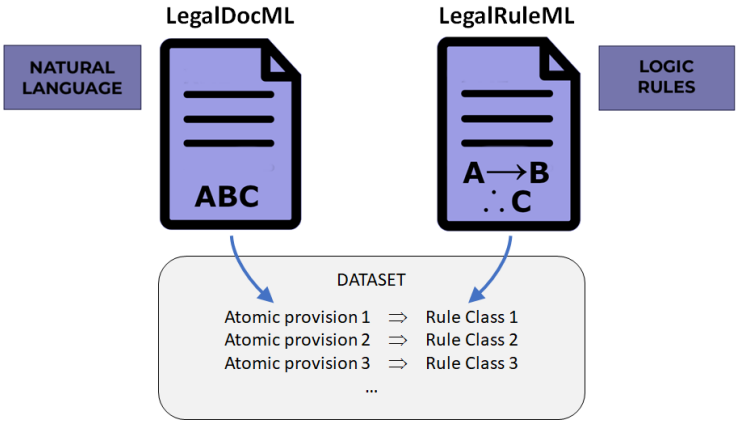


Fig. 1. Knowledge extraction from LegalDocML and LegalRuleML. Note that each extracted instance refers to an atomic normative provision (generally contained in paragraphs or points), and may sometimes consist of more than one sentence.

Specifically, we created a dataset of 707 atomic legal provisions out of the 966 LegalRuleML representations (271 obligations, 76 permissions, and 619 constitutive rules) in the DAPRECO-KB. By following the pointers to the LegalDocML IDs, we were able to reconstruct the exact natural language target, even when the provisions were split into lists. By combining the structural information from LegalDocML and the logical content from LegalRuleML we extracted 707 labelled legal provisions in total. The labels of these sentences are the same as those in the DAPRECO-KR with the addition of a "non-rule" category. We abbreviated "obligationRule", "permissionRule", "constitutiveRule" in "obligation", "permission" and "constitutive" respectively.

The class "obligation" is referred to those sentences which have at least one obligation rule in their related formulae. The class "permission" is referred to those sentences which have at least one permission rule in their related formulae. The class "constitutive" is referred to those sentences which just constitutive rules in their related formulae. Constitutive rules are used to trigger specific inferences for the modeled rules and are distinct from obligations or permissions in that they do not convey information about deontic modalities. Finally, we also considered a class "non-rule" which is referred to all sentences which have no legal rule at all, and "non-deontic" which is referred to all sentences which does not contain neither obligations nor permissions (they may still contain constitutive rules, though)[3].

These labels allow for four different experimental settings, as shown in Table 1, which provide different levels of granularity.

Table 1. Number of instances per class per scenario.

	Classes	**Instances**
Scenario 1	rule	260
	non-rule	447

	Classes	**Instances**
Scenario 2	deontic	204
	non-deontic	503

	Classes	**Instances**
Scenario 3	obligation	156
	permission	44
	non-deontic	503

	Classes	**Instances**
Scenario 4	obligation	156
	permission	44
	constitutive	56
	non-rule	447

3.2 Classification method

The idea underneath the methodology for the second objective is that Transfer Learning methods can have good performances even with small datasets.

Transfer Learning generally consists in the use of neural architectures pre-trained on huge amounts of data. These neural architecture are sometimes pre-trained on tasks which are designed to "force" the neural architecture to forecast some aspect of language along with connections between words.

On the one hand, the results of this process of pre-training a neural architecture over a huge amount of data generates language models which can achieve remarkable results in many NLP tasks; on the other hand, the "knowledge" acquired by this pre-trained neural architectures during the training, can be "transferred" (hence the name "Transfer Learning") on downstream, more specific, tasks, which can even use small datasets.

[3] For the multi-classifications (i.e. Scenario 3 and 4) four statements have been removed, since the classes "obligation" and "permission" overlapped.

As stated earlier, we used BERT, DistilBERT, LegalBERT, and GPT-3 to fine-tune their pre-trained language models on the 707 labelled legal provisions extracted from the DAPRECO-KB as explained in the previous section.

To fine-tune the approach with GTP-3, we engineered two different simple prompts to give GPT-3 the necessary instructions for classifying the atomic provisions' classes. Our first prompt has the following template:

prompt: "[ATOMIC LEGAL PROVISION]\n\nThe previous text is a ->"
completion: "[CLASS AS NUMBER]"

Where "[ATOMIC LEGAL PROVISION]" is the single atomic provision extracted from LegalDocML and LegalRuleML, "->" is our classification marker, and "\n" stands for a new line. The completion of this prompt is the class represented as a number (in the case of scenario 4, numbers 0, 1, 2, 3 stand for "none", "obligation", "permission" and "constitutive" respectively).

The second prompt has the following template:

prompt: "[ATOMIC LEGAL PROVISION]\n\nThe previous text is a ->"
completion: "[CLASS NAME]"

In this second prompt, the completion of this prompt is the class of the atomic legal provision represented as words (not as numbers).

An example of legal provision marked using prompt 1 and 2 is the following:


```
{"prompt":"The data subject shall have the right to obtain from the controller confirmation as to whether or not
personal data concerning him or her are being processed, and, where that is the case, access to the personal data
and the following information the recipients or categories of recipient to whom the personal data have been or
will be disclosed, in particular recipients in third countries or international organisations;\n\nThe previous
text is a ->","completion":" 1"}
```

```
{"prompt":"The data subject shall have the right to obtain from the controller confirmation as to whether or not
personal data concerning him or her are being processed, and, where that is the case, access to the personal data
and the following information the recipients or categories of recipient to whom the personal data have been or
will be disclosed, in particular recipients in third countries or international organisations;\n\nThe previous
text is a ->","completion":" obligation"}
```


Fig. 2. Example of legal provision with prompt 1 (top) and prompt 2 (bottom)

4 Data

In this study, we employed a dataset containing a total amount of 707 atomic normative provisions[4] extracted from the European GDPR (General Data Protection Regulation) through the DAPRECO-KB as explained in subsection 3.1.

[4] For the selection of the provisions, we excluded preamble and conclusion from the main legal document of the GDPR, thus keeping just the provisions within the body of the GDPR. These provisions are generally paragraphs or list points, and may sometimes consist of more than one sentence. The dataset is available at: https://gitlab.com/daviligade/gdpr_akn_legalruleml

It is important to remark that this combination of LegalDocML and LegalRuleML also facilitates the reconstruction of the exact target, in terms of natural language, where each provision is located. For example, many obligations of legal texts are split into lists, and LegalDocML is crucial to reconstruct those pieces of natural language into a unique piece of natural language. For example, Article 5 of the GDPR[5] states:

> *Article 5*
> **Principles relating to processing of personal data**
> 1. Personal data shall be:
> (a) processed lawfully, fairly and in a transparent manner in relation to the data subject ('lawfulness, fairness and transparency'); (b) collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes; further processing for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes shall, in accordance with Article 89(1), not be considered to be incompatible with the initial purposes ('purpose limitation');
> (c) adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed ('data minimisation'); *[...]*

As can be seen in the text above, *paragraph 1* of *Article 5* is a list composed of an introductory part ("Personal data shall be:") and different points. For simplicity, we only reported the first three points of *paragraph 1*.From the point of view of the natural language, each deontic sentence is split between the introductory part (which contains the main deontic verb "shall") and the text of each point. While the introductory part contains the main deontic verb, the actual deontic information is contained within each point. The LegalDocML formalization for *point a* is the following:

```
<article eId="art_5">
  <num>Article 5</num>
  <heading eId="art_5__heading">Principles relating to processing of
                                          personal data</heading>
  <paragraph eId="art_5__para_1">
    <num>1.</num>
    <list eId="art_5    __para_1__content__list_1">
      <intro><p>Personal data shall be:</p></intro>
      <point eId="art_5__para_1__content__list_1__point_a">
        <num>(a)</num>
         <content><p> processed lawfully, fairly and in a
             transparent [...]</p></content>
       </point>
  [...]
```

[5] https://eur-lex.europa.eu/eli/reg/2016/679/oj#d1e1807-1-1.

In the DAPRECO-KB, a series of `<LegalReference>` elements can be found, which contain the structural portion where the deontic formulas are located, referenced by using the LegalDocML naming convention[6]. For example, the reference of the above mentioned *point a* is encoded in the D-KB as follows:

```
<LegalReference refersTo="gdprC2A5P1p1ref"
            refID="GDPR:art_5__para_1__content__list_1__point_a">
```

in which the `refersTo` attribute indicates the internal ID of the reference, and the "refID" attribute indicates the external ID of the reference using the LegalDocML naming convention. The prefix "GDPR" is the LegalDocML uri of the GDPR, i.e., `/akn/eu/act/regulation/2018-05-25/eng@2018-05-25/!main#`.

In turn, this `<LegalReference>` element is then associated to its target group of logical statements, which collects the group of logical formulas related to this legal reference (so, in this case, related to *point a* of the first paragraph of *Article 5*). Such association is modelled as follows:

```
<Association>
    <appliesSource keyref="#gdprC2A5P1p1ref">
    <toTarget keyref="#statements1">
</Association>
```

Where the attribute `keyref` of the target connects the source to the collection of statements whose `key` attribute is `statements1`:

```
<Statements key="statements1">
  <ConstitutiveStatement key="statements1Formula1">
    <Rule closure="universal">
      <if>[...]</if>
      <then>[...]</then>
    </Rule>
  </ConstitutiveStatement>
  <ConstitutiveStatement key="statements1Formula2">
      <if>[...]</if>
      <then>[...]</then>
    </Rule>
  </ConstitutiveStatement>
</Statements>
```

It is important to underline that each natural language statement can have multiple formulas in the logical sphere. For this reason, the element `<Statements>` here shows a collection of two logical formulas.

To finally associate the portion of natural language extracted from LegalDocML to a class related to the logical sphere (i.e. the deontic class), one must look at the `<Context>` elements which are related to the two formulas we found.

[6] https://docs.oasis-open.org/legaldocml/akn-nc/v1.0/csprd01/akn-nc-v1.0-csprd01.html.

```
<Context key="context_1" type="rioOnto:obligationRule">
  <inScope keyref="#statements1Formula1"/>
</Context>
<Context key="context_3" type="rioOnto:constitutiveRule">
  <inScope keyref="#statements1Formula2">
</Context>
```

As can be seen from the text above, the first formula (which is called here `statements1Formula1`) is associated with the ontological class `obligationRule`, while the second formula (which is called `statements1Formula2`) is associated with the ontological class `constitutiveRule`. This means that the piece of natural language expressed in *point a* of the first paragraph of *Art. 5* of the GDPR contains, at the logical level, a constitutive rule and an obligation rule.

Figure 3 shows the full series of steps from the natural language sphere (located in the LegalDocML) to the logical sphere (i.e. the LegalRuleML formalization) where the deontic classes are located. The figure explains step-by-step how the combination of LegalDocML and LegalRuleML helped us in the extraction of annotated labelled data.

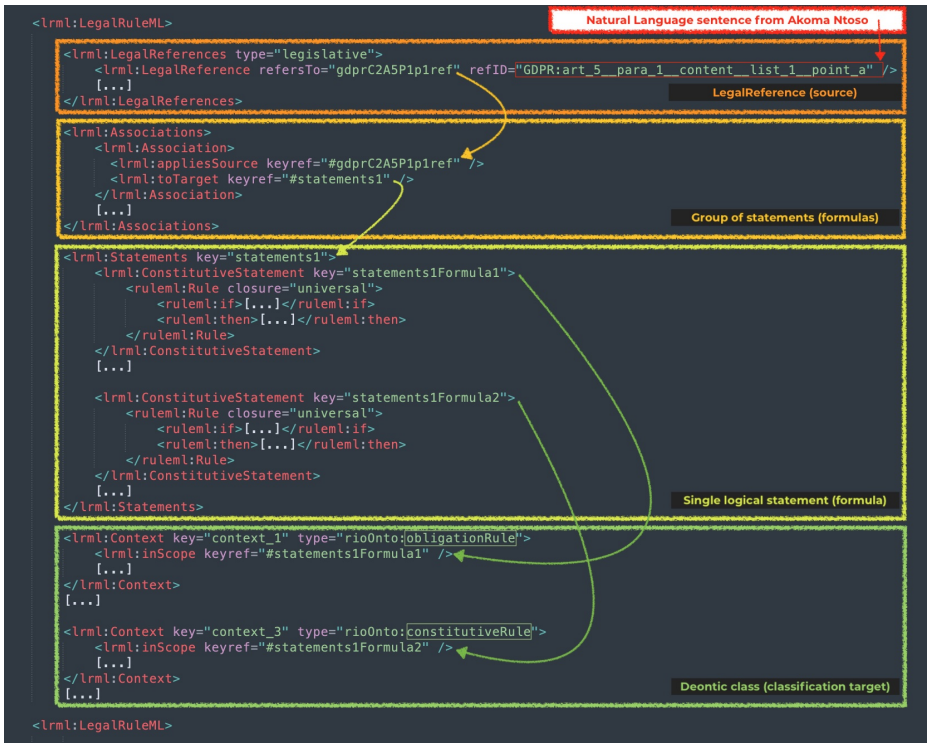


Fig. 3. Class extraction process from the DAPRECO-KB.

5 Experiment settings and results

As far as the experimental settings are concerned, the dataset was divided into training and validation sets, with a 80/20 split. Moreover, as engine for GPT-3 we employed Ada, the standard choice for classification tasks. Also, we noticed empirically that GPT-3's Ada outperforms Davinci in simple classification tasks, while Davinci is more appropriate in generative tasks.

Table 2 reports the result of the three BERT-based pre-trained models and the two ones based on GPT-3, i.e., GPT-3 trained on the two prompts explained in subsection 3.2 above (shown as "p.1" and "p.2" in the table); these have been fine-tuned after 4 epochs. As far as the hyper-parameters are concerned, we set the learning rate multiplier at 0.1, the prompt loss weight at 0.01 and the batch size at 1. The final results on the validation set are reported in Table 2, where it can be seen that all fine-tuned GPT-3 models outperforms the BERT-based ones, both in terms of F1 scores (left values) and in terms of accuracy (right values). Figure 4 shows a graph with accuracy and F1 scores for the two GPT-3 fine-tuned models, processed both via Davinci and Ada; the figure shows that Ada outperforms Davinci in the tasks considered in this paper, as stated earlier.

Table 2. Results for the four classification scenarios. Evaluation metrics: F1-score (left) and Accuracy (right). For the multiclass scenarios 3 and 4 we used weighted F1. RI indicated the relative improvement in decimal points compared to the best baseline.

	models based on BERT			models based on GPT-3		
	BERT	**DistilBERT**	**LegalBERT**	**GPT-3(p.1)**	**GPT-3(p.2)**	RI
Scenario1	.86/.86	.88/.88	.82/.82	.90/.94	.91/.94	+3/+6
Scenario2	.88/.88	.92/.92	.88/.88	.90/.95	.93/.96	+1/+4
Scenario3	.88/.87	.84/.83	.85/.84	.94/.94	.93/.93	+6/+7
Scenario4	.78/.75	.80/.78	.81/.76	.91/.91	.93/.93	+12/+15

6 Conclusion and Future works

This paper showed that autoregressive generative pre-trained language models such as GPT-3 can outperform auto-encoding pre-trained language models such as BERT for the task of legal rule classification.

We consider our results to be general enough for that task as we run experiments on *four* different multiclass classification scenarios, involving obligations, permissions, constitutive rules and "non-rules" (i.e., legal provisions which do not contain any kind of rule).

Importantly, our work shows how Hybrid AI approaches can successfully combine symbolic and sub-symbolic Artificial Intelligence. The novelty and the power of generative AI methodologies jointly with the combined use of LegalDocML and LegalRuleML are two major contributions of this study, along with the design of the experimental settings in four different classification scenarios using two different prompt strategies.

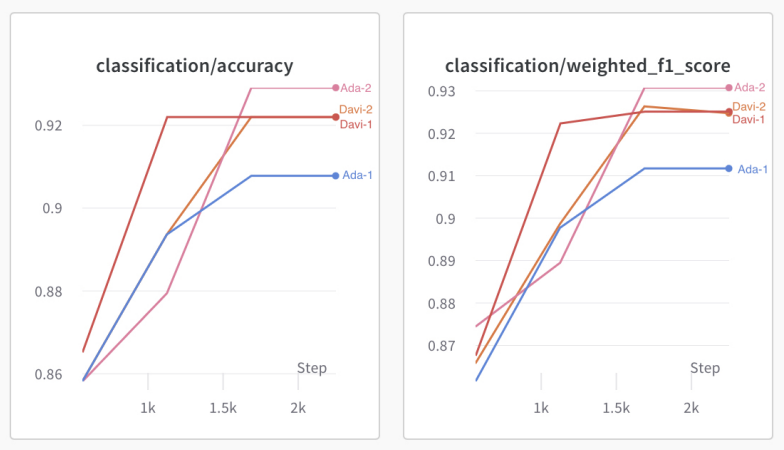


Fig. 4. accuracy and F1 scores for the GPT-3 fine-tuned models.

It is likewise worth noticing that legal XML formats such as LegalDocML and LegalRuleML are usually written and validated by legal experts, which can "inject" specific domain expertise into the pre-trained language model. In other words, the extraction of data from documents in these XML standards for the legal domain can arguably offer a more convenient and robust solution compared to the use of general-purpose datasets.

This Hybrid AI approach shows the potential of combining top-down, i.e., knowledge-driven, approaches with bottom-up, i.e., data-driven, methods. To the best of our knowledge, this paper represents the first attempt in the AI&Law literature to fine-tune GPT-3, currently the most powerful and discussed pre-trained language model, on legal data.

However, the present work only represents the first step of our research journey. More experiments are needed to confirm this trends on other legal datasets.

In the future, we plan to explore GPT-3's ability to deal with more complex and granular tasks of legal rule classification, e.g., semantic role labelling [13]. Another important direction might be that of creating expert systems capable to automatically translate textual norms into logical representations and checking compliance accordingly, similarly to what has been recently done in [22], which also uses the DAPRECO-KB as reference repository.

References

1. Antoniou, G., Atkinson, K., Baryannis, G., Batsakis, S., Di Caro, L., Governatori, G., Robaldo, L., Siragusa, G., I., T.: Large-scale legal reasoning with rules and databases. Journal of Applied Logics - IfCoLog Journal **8**(4) (2021)
2. Ashley, K.D.: Artificial intelligence and legal analytics: new tools for law practice in the digital age. Cambridge University Press (2017)

3. Athan, T., Boley, H., Governatori, G., Palmirani, M., Paschke, A., Wyner, A.: Oasis legalruleml. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law. pp. 3–12 (2013)
4. Boella, G., Di Caro, L., Humphreys, L., Robaldo, L., Rossi, R., van der Torre, L.: Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law. Artificial Intelligence and Law **4** (2016)
5. Boella, G., Caro, L.D., Rispoli, D., Robaldo, L.: A system for classifying multi-label text into EuroVoc. In: Proc. of the International Conference on Artificial Intelligence and Law (ICAIL). pp. 239–240. ACM (2013)
6. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)
7. Chalkidis, I., Androutsopoulos, I., Michos, A.: Obligation and prohibition extraction using hierarchical rnns. arXiv preprint arXiv:1805.03871 (2018)
8. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: Legal-bert: The muppets straight out of law school. arXiv preprint arXiv:2010.02559 (2020)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Dragoni, M., Villata, S., Rizzi, W., Governatori, G.: Combining natural language processing approaches for rule extraction from legal documents. In: Pagallo, U., Palmirani, M., Casanovas, P., Sartor, G., Villata, S. (eds.) AI Approaches to the Complexity of Legal Systems. pp. 287–300. Springer International Publishing, Cham (2018)
11. Gao, X., Singh, M.P.: Extracting normative relationships from business contracts. In: Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. pp. 101–108 (2014)
12. Gomez-Perez, J.M., Denaux, R., Garcia-Silva, A.: Hybrid Natural Language Processing: An Introduction, pp. 3–6. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-44830-1{_}1, https://doi.org/10.1007/978-3-030-44830-1_1
13. Humphreys, L., Boella, G., van der Torre, L., Robaldo, L., Di Caro, L., Ghanavati, S., Muthuri, R.: Populating legal ontologies using semantic role labeling. Artificial Intelligence and Law **29**, 171–211 (2021)
14. Joshi, V., Anish, P.R., Ghaisas, S.: Domain adaptation for an automated classification of deontic modalities in software engineering contracts. In: Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. pp. 1275–1280 (2021)
15. Kiyavitskaya, N., Zeni, N., Breaux, T.D., Antón, A.I., Cordy, J.R., Mich, L., Mylopoulos, J.: Automating the extraction of rights and obligations for regulatory compliance. In: International Conference on Conceptual Modeling. pp. 154–168. Springer (2008)
16. Liga, D.: Hybrid artificial intelligence to extract patterns and rules from argumentative and legal texts (2022)

17. Liga, D., Palmirani, M.: Deontic sentence classification using tree kernel classifiers. In: Intelligent Systems and Applications: Proceedings of the 2022 Intelligent Systems Conference (IntelliSys) Volume 1. pp. 54–73. Springer International Publishing Cham (2022)
18. Liga, D., Palmirani, M.: Transfer learning for deontic rule classification: the case study of the gdpr. In: International conference on legal knowledge and information systems (2022)
19. de Maat, E., Winkels, R.: Automatic classification of sentences in dutch laws. In: Legal Knowledge and Information Systems, pp. 207–216. IOS Press (2008)
20. Nanda, R., Caro, L.D., Boella, G., Konstantinov, H., Tyankov, T., Traykov, D., Hristov, H., Costamagna, F., Humphreys, L., Robaldo, L., Romano, M.: A unifying similarity measure for automated identification of national implementations of european union directives. In: Proc. of the International Conference on Artificial Intelligence and Law (ICAIL). ACM (2017)
21. Neill, J.O., Buitelaar, P., Robin, C., Brien, L.O.: Classifying sentential modality in legal language: a use case in financial regulations, acts and directives. In: Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law. pp. 159–168 (2017)
22. Nguyen, M., Nguyen, T., Tran, V., Nguyen, H., Nguyen, L., Satoh, K.: Learning to map the GDPR to logic representation on DAPRECO-KB. In: ACIIDS (1). Lecture Notes in Computer Science, vol. 13757, pp. 442–454. Springer (2022)
23. Palmirani, M., Vitali, F.: Akoma-ntoso for legal documents. In: Legislative XML for the semantic Web, pp. 75–100. Springer (2011)
24. Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018)
25. Robaldo, L., Bartolini, C., Palmirani, M., Rossi, A., Martoni, M., Lenzini, G.: Formalizing gdpr provisions in reified i/o logic: the dapreco knowledge base. Journal of Logic, Language and Information **29**(4), 401–449 (2020)
26. Robaldo, L., Villata, S., Wyner, A., Grabmair, M.: Introduction for artificial intelligence and law: special issue "natural language processing for legal texts". Artificial Intelligence and Law **27**(2), 113–115 (2019)
27. Rodríguez-Doncel, V., Palmirani, M., Araszkiewicz, M., Casanovas, P., Pagallo, U., Sartor, G.: Introduction: A hybrid regulatory framework and technical architecture for a human-centered and explainable ai. In: AICOL XI-XII, pp. 1–11. Springer (2020)
28. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
29. Shaghaghian, S., Feng, L.Y., Jafarpour, B., Pogrebnyakov, N.: Customizing contextualized language models for legal document reviews. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 2139–2148. IEEE (2020)
30. Sun, X., Robaldo, L.: On the complexity of input/output logic. The Journal of Applied Logic **25**, 69–88 (2017)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2017), https://arxiv.org/pdf/1706.03762.pdf
32. Waltl, B., Muhr, J., Glaser, I., Bonczek, G., Scepankova, E., Matthes, F.: Classifying legal norms with active machine learning. In: Wyner, A.Z., Casini, G. (eds.) Legal Knowledge and Information Systems JURIX 2017. Frontiers in Artificial Intelligence and Applications, vol. 302. IOS Press (2017)
33. Wyner, A., Peters, W.: On rule extraction from regulations. In: Legal Knowledge and Information Systems, pp. 113–122. IOS Press (2011)

OntoVAT, an ontology for knowledge extraction in VAT-related judgments*

Davide Liga[1][0000−0003−1124−0299], Alessia Fidelangeli[2], and Réka Markovich[1]

[1] University of Luxembourg, Esch-sur-Alzette, Luxembourg
{davide.liga,reka.markovich}@uni.lu
[2] Alma Mater Studiorum - University of Bologna, Bologna, Italy
alessia.fidelangeli2@unibo.it

Abstract. In this work, we introduce OntoVAT, a multilingual ontology designed for knowledge extraction in VAT-related legal judgments. To the best of our knowledge, this is the first comprehensive ontology in the field of VAT (Value-Added Tax). The main aims of this ontology are to capture the key concepts involved in the European VAT domain and to provide an extendible and reusable knowledge representation to facilitate the automated extraction or detection of VAT-related concepts in legal judgments. This ontology can also facilitate many other tasks of Artificial Intelligence and Law (AI&Law), e.g., legal knowledge extraction, keyword extraction, topic modeling, and semantic relations extraction. OntoVAT is created using OWL as the basic format of representation, with a SKOS lexicalization. We present here a first version of the ontological patterns and relations of the ontology, which we release in three different languages and which is the result of an ongoing effort between computer scientists and domain experts.

Keywords: Legal Knowledge Representation · Ontology · VAT · AI&Law.

1 Introduction

The field of Artificial Intelligence and Law (AI&Law) has seen a huge growth in recent years, with a range of applications being developed to assist legal professionals, improve access to justice, and facilitate the functioning of legal systems. One critical aspect in the development of AI&Law applications is the representation and management of knowledge, which is essential for ensuring that systems can operate effectively and deliver accurate results. Ontologies, which are formal representations of a specific domain's knowledge, can contribute to achieving this objective, and can have a crucial role in combination with

* This works has been supported by the Analytics for Decision of Legal Cases (ADELE), founded by the European Union's Justice Programme (grant agreement No. 101007420); Davide Liga was supported by the project INDIGO, which is financially supported by the NORFACE Joint Research Programme on Democratic Governance in a Turbulent Age and co-funded by AEI, AKA, DFG and FNR and the European Commission through Horizon 2020 under grant agreement No 822166

non-symbolic and sub-symbolic AI methods [10]. In fact, ontologies are crucial tools for the advancement of the field of AI&Law, since they provide a way to accurately represent complex symbolic knowledge in machine-readable format, while preserving the advantages coming from being modular and inter-operable components. In this work, we propose a first version of OntoVAT, an ontology designed for knowledge extraction from legal judgments related to Value Added Tax (VAT). The main aims of this ontology are to capture the key concepts involved in the European VAT domain and to provide an extendible and reusable knowledge representation for extracting VAT-related concepts for the analysis of judicial decisions or, more generally, for the analysis of judgments. These kinds of ontology can facilitate tasks such as the retrieval of keywords, topic modeling, the extraction of semantic relations, etc.

In the next sections, we will describe the few related works and our own contributions (see Section 2), the methodology we adopted (see Section 3), and the current structure of the ontology (see Section 4). Finally, in the last part of the work we will provide some suggestions for future developments in the field (see Section 5).

2 Related Works and Contributions

Ontologies are important tools in the field of AI&Law [11], and have been used in various contexts such as the modeling of privacy law [9] or the recent Artificial Intelligence Act [2]. Nonetheless, there are no attempts to build a comprehensive ontology related to Value-Added Tax (VAT). To the best of our knowledge, the only attempt to build an ontology in this field dates back to 20 years ago [7] [13], when Karremans et al. pursued to describe a few potential core ontological concepts related to VAT. However, their work was more dedicated to showing the obstacles related to the design of complex multilingual ontologies (where culture-specific or language-specific elements can create constraints or limitations during the design of the ontology) than aimed at creating a complete VAT ontology. The authors' proposal was limited to a few interesting conceptual suggestions for the development of a potential VAT ontology.

This absence of related works is probably due to the difficulty in reconstructing such a complex and articulated legal (and conceptual) domain. Indeed, the creation of an ontology in the field of VAT entails many critical issues: (1) while most VAT concepts are harmonized at the European level, others are regulated (or even mentioned) only at the national level; (2) the VAT regulation relies on the use of concepts belonging to other domains of law (such as civil law or commercial law) or common language concepts which are employed with a particular meaning in the field of VAT; (3) many VAT concepts are not defined by the VAT Directive or national legislation, but by the case law of the Court of Justice of the European Union (CJEU). Thus, on the one hand, the modeling of VAT concepts requires an analysis on multiple levels, considering: European legislation, case law, and national implementations. On the other hand, it requires an analysis of concepts belonging to several fields of law, as well as to

common language. We decided to build an ontology at an intermediate-low layer of abstraction while committing it to already existing upper ontologies. In this regard, there are already many other upper ontologies designed to represent higher levels of abstraction, including the Legal Knowledge Interchange Format (LKIF), an upper ontology designed for legal knowledge [6]).

Another important aspect behind the design of OntoVAT is that it has an applicative intended use. It has been designed with the purpose of capturing the concepts which might be crucial in the legal reasoning of VAT-related judgments and especially with decisions concerning taxable/exempt VAT transactions. Hence, we had to focus both on relatively abstract concepts such as "transaction" or "place", which were frequently mentioned in the above-mentioned decisions, as well as on more specific concepts beloning to the domain of VAT (such as the concepts of "exemption" or "supply of goods"), or to specific areas of knowledge (for example "vessels" or "human blood"). The above-mentioned challenges are related to the difficulty of building an ontology capable of being expressive and representing such a large number of layers of abstractions belonging to different conceptual areas. A further challenge was to ensure the consistency of the resulting model from a formal point of view. For this reason, we decided to create this ontology in OWL format, so as to provide the scientific community with a first formal ontology, on which to explore automated reasoning experiments. Here, we present this first version of OntoVAT as a multilingual ontology (implemented in English, Italian, and Bulgarian) which is both consistent from a formal point of view and tailored to a specific applicative goal, namely modeling the most crucial concepts in VAT-related legal judgments.

3 Methodology

For the creation of OntoVAT, we were inspired by [9], which adopted a methodology to minimise the difficulties for legal operators to define a legal ontology.

We followed a top-down approach applied on legal sources and made more robust by the partial reuse of pre-existing ontology patterns [5]. Our results are evaluated by using foundational ontologies (in particular LKIF [6], DOLCE [3] and DUL [1]), and we followed the principles in the OntoClean [4] method, according to which each ontological concept can be evaluated based on three meta-properties:

1. "identity" (making sure that a class uniquely identifiable)
2. "unity" (making sure that instances of a class form cohesive and meaningful wholes)
3. "rigidity" (whether a property is essential to the instances of a class or if it can change over time)

Our validation involved a strongly interdisciplinary group, mostly composed of computer scientists, lawyers, and philosophers, which allowed an integrated expertise coming from different disciplines.

We can summarise our approach in the following steps:

(i) a group of legal experts selected nearly 500 legal judgements related to the domain of VAT in Italian and Bulgarian;
(ii) the judgements were analyzed and the portions of text related with the judges' motivations were annotated;
(iii) Italian and Bulgarian legal experts analysed the most important concepts mentioned in the judgements, checking these concepts against their respective statutory backgrounds;
(iv) our technical team received the selected concepts and portions of text from the legal experts to map them into the ontology;
(v) for each element of the ontology our legal experts provided a range of linguistic variations/synonyms, a definition, the most common examples instantiating that concept, the most common related terms, and any relevant normative references related to the concept;
(vi) the gathered results were validated by the legal team that returned them to the technical team who implemented the new information in the ontology;
(vii) the steps from (iii) to (vi) were iterated several times to refine the ontology;

We are also in the process of implementing an algorithm which uses the OntoVAT to determine whether an ontological concept is relevant on judgements related to VAT, i.e. if a specific decision deals with one or more of the ontological concepts. This process can be summarised as follows:

1. legal experts were asked to select from OntoVAT the ontological concepts which are considered more relevant in the decisions of judges;
2. considering the concepts selected in the previous step, legal experts were asked to manually annotate nearly 70% of the judgements by including the information of whether each selected concept is relevant in each judgement by associating a binary value, where 0 means "non relevant" and 1 means "relevant" (the concept is considered relevant if the court's decision concerns that concept from the substantial point of view);
3. an algorithm designed by the technical team encodes the information contained in the ontology to predict whether or not a concept is relevant (comparing the results with the gold standard defined in the previous step);

We are currently in the process of completing step 2 and implementing step 3. Our preliminary results shows that by using OntoVAT we can catch the most important relevant concepts in the judicial decisions.

This methodology can be generalized and applied to different domains (and can be easily extended to other languages). For example, we employed the same approach for the development of another ontology, PaTrOnto, related to the domain of patents and trademarks [8]. The main difference between PaTrOnto and OntoVAT is related to the above-mentioned step (iii), since the statutory backgrounds for the field patents and trademarks is completely different, also in terms of harmonisation at the European level.

4 The design of OntoVAT

4.1 Core concepts

It is worth mentioning that the ongoing effort behind this work is the result of the cooperation between computer scientists and legal experts in the VAT domain. Regarding the design of OntoVAT, we proceeded by taking into account different sources of information. First of all, we considered the European VAT Directive, which is the main legal source at the European level. The Directive provides a harmonized and coherent perspective on the ontological concepts of the VAT domain and it is compatible with our target of creating a multilingual VAT ontology, as it is available in all the official languages of the European Union. Moreover, we also considered another source of information, namely the case law of the CJEU, which we found particular useful to find key concepts which were not defined by the Directive. Finally, we tried to model the key ontological concepts with an even more concrete source of information, namely the (above-mentioned) dataset of VAT-related judgments adopted by national courts. More specifically, we analyzed which concepts were particularly important in the legal reasoning of national judges, and how these concepts were employed by them. Thus, while the Directive was the fundamental starting point of the work, this was complemented by further research aimed at identifying the concepts that were actually relevant in real cases decided by national courts or by the CJEU. Therefore, one of the first challenges was to reconcile these two aspects (i.e., the more abstract normative dimension and the more concrete dimension of judicial cases).

Inspired by the first articles of our first source (i.e., the EU VAT Directive), we decided to put at the center of our ontology the concept of "Transaction", around which we added all the other concepts. The core ontological concepts are shown in Figure 1.

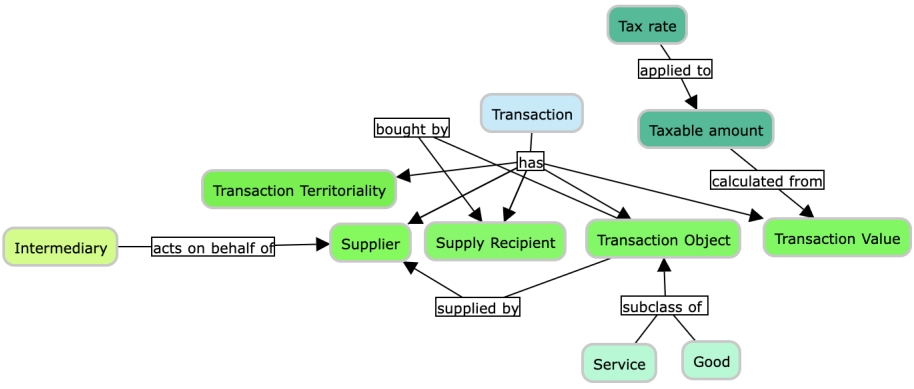


Fig. 1. The core elements of OntoVAT.

Thus, the idea is that any transaction which may (or may not) be subject to VAT will have some agents involved (a supplier, a recipient, and sometimes intermediaries), an exchanged object (generally, a service or a good), and an exchanged value from which the taxable amount is calculated. We also added the concept of territoriality, as it has consequences on the fact that the transaction is actually taxed. Starting from these core ontological concepts, we then further developed the ontology by extending their modeling. For example, a challenging step during the design of the ontology was related to the modeling of the *objective profiles* and the *subjective profiles* of the transaction, i.e., which people are subject to VAT according to the European VAT Directive (which people are taxable persons), and what kind of transactions and transactions objects (e.g., types of goods and services) are relevant for the judges to take their decisions. Furthermore, we included the concept of "Exemption" and "Right to deduction", modeling also the relation with the concept of "VAT Chargeable Event", since we realized that these concepts were very relevant in our dataset of national decisions.

4.2 OntoVAT details and lexicalisation

The ontology is currently composed of 129 concepts (i.e., OWL classes) and 36 properties (relationships between classes). A more exhaustive numerical description is reported in Table 1.

Element	Quantity
Number of classes	122
Number of properties	28
Number of datatype properties	8
Number of transitive properties	0
Number of disjoint class pairs	578
Number of subclass relations	101

Table 1. OntoVAT's statistics.

OntoVAT is a multilingual OWL ontology enriched with a SKOS lexicalisation and implemented in English, Italian and Bulgarian. This OWL+SKOS multilingual implementation has been implemented using VocBench 3 [12] and is a powerful approach to mitigate the issue of semantic non-uniformity in multilingualism, which has been pointed out in previous research [7]. Thanks to the use of SKOS, each ontological concept (i.e. each OWL class) is enriched with some specific properties which are incorporated in the SKOS data model, namely:

- skos:definition
- skos:scopeNote
- skos:altLabel

- skos:hiddenLabel
- skos:example

The addition of these properties to each ontological concept (in English, Italian and Bulgarian) facilitates the integration of crucial information within the ontology, making OntoVAT particularly expressive and powerful. In particular, **skos:definition** contains the definition of each single OWL class (i.e., the definition of each single concept). In **skos:scopeNote**, we added relevant specifications about the skos:definition field (whenever was necessary to further specify the interpretative angle of the chosen definition). Furthermore, scopeNotes also contain all relevant normative references (if any) describing the concept. We also added any relevant synonyms in the three different languages as **skos:altLabel** properties. In **skos:example**, we added some examples of the concept (which might look like further potential subclasses of the concept). Finally, the property **skos:hiddenLabel** is used to store terms in natural language which might signal the presence of the concept in the text (this can be useful for any application layers built on top of OntoVAT).

As mentioned before, we built OntoVAT using concepts taken from the European VAT Directive to grant a coherent and harmonic conceptual framework. Therefore all concepts are already designed to be appropriate for both Italy and Bulgaria. In fact, Italy and Bulgaria must grant the uniform application of European law.

In most cases, the semantic meaning of concepts is therefore harmonic between Italy and Bulgaria. In these cases, for each OWL class, a skos:definition is just provided in English and translated into Italian and Bulgarian with no adjustments. However, in few cases, definitions of concepts (i.e., their semantic meaning) vary at national level. In these situations, priority was given to national definitions, therefore the skos:definition in Bulgarian/Italian will not be just a translation from English, instead it will be a different definition (coherent with the national legislation). Moreover, whenever further specifications are needed to explain the scope of the concepts' meaning (at Bulgarian, Italian, and European level), we employed a skos:scopeNote property in Bulgarian/Italian/English.

Lastly, since national legislation may have alternative terms for referring to the Directive's concepts, we handled alternative terms as synonyms (skos:altLabel) in Italian/Bulgarian. For the time being we did not introduce any country-specific class, as our goal was to develop a common ontology which could be used by both Italian and Bulgarian judges. Moreover, the creation of a common ontology may be useful in developing a common conceptual framework that promotes the uniform application of EU law in a harmonised field. In the future, we will consider extending our ontology by adding specific classes based on concepts which are used by the legislator in national implementation. This might be useful for national judges, who might be more familiar with different country-specific concepts.

Hence, we handle the issue of multilinguality by specialising the skos properties skos:definitions, skos:scopeNotes and skos:altLabels whenever needed, with-

out affecting the coherence of the ontological concepts or their relations (Figure 2 shows an example of how multilinguality is handled for a specific concept/class).

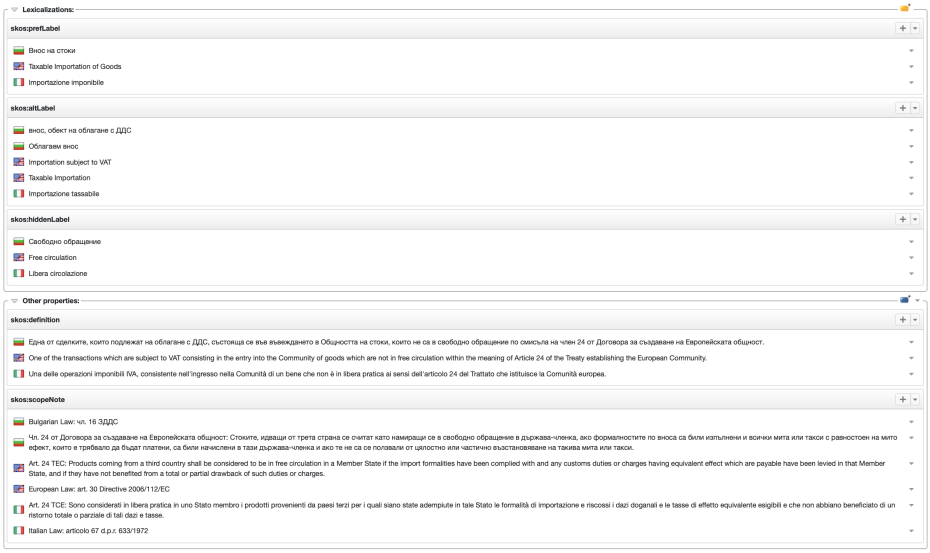


Fig. 2. An example of multilingual lexicalisation, related to the OWL class (i.e. the concept) "Taxable Importation of Goods".

We carefully assigned a definition to each concept by giving priority to definitions coming from the domain-specific legislative sources, whenever the concept exists in that domain. If the concept is not mentioned neither in the national nor in the European legislative sources, we searched for a definition in the case law of the Court of Justice of the European Union (CJEU). If the concept is not defined neither in the legislation nor in the case law of the CJEU, as it frequently happens for "factual concepts", it is defined following a simple description based on legal encyclopedias or dictionaries. In this way, we made sure that the definition of each concept coherently anchored to the legals sources.

4.3 Commitment and scope

Figure 3 shows a simplified conceptual map that gives a clearer understanding of the formal structure of the ontology, showing most ontological classes and properties which can be found in the OWL ontology[3]. In this map, one can see the previously mentioned core elements having the class "Transaction" as

[3] Relations such as "has" connecting to a target concept are represented in OWL as "hasTargetConcept", while relations such as "can be" are translated in OWL as datatype properties with a boolean value.

central concept, as we previously described in Figure 1. To make the picture more readable, the classes “Supplier”, “Good” and “Service” have been duplicated and expanded at the bottom of the map and some classes have been omitted. Please see Figure 4 for the complete hierarchies of classes and properties in OntoVAT.

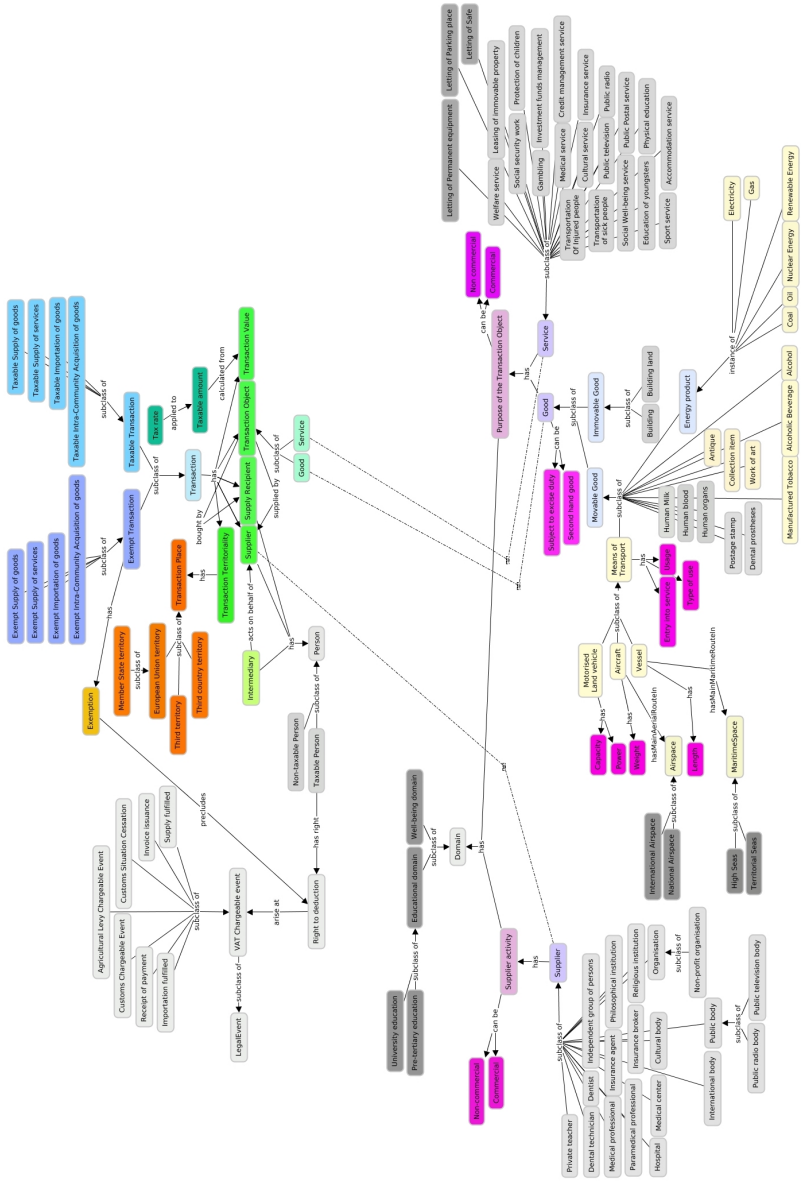


Fig. 3. Simplified map of the main concepts and relations in OntoVAT.

.

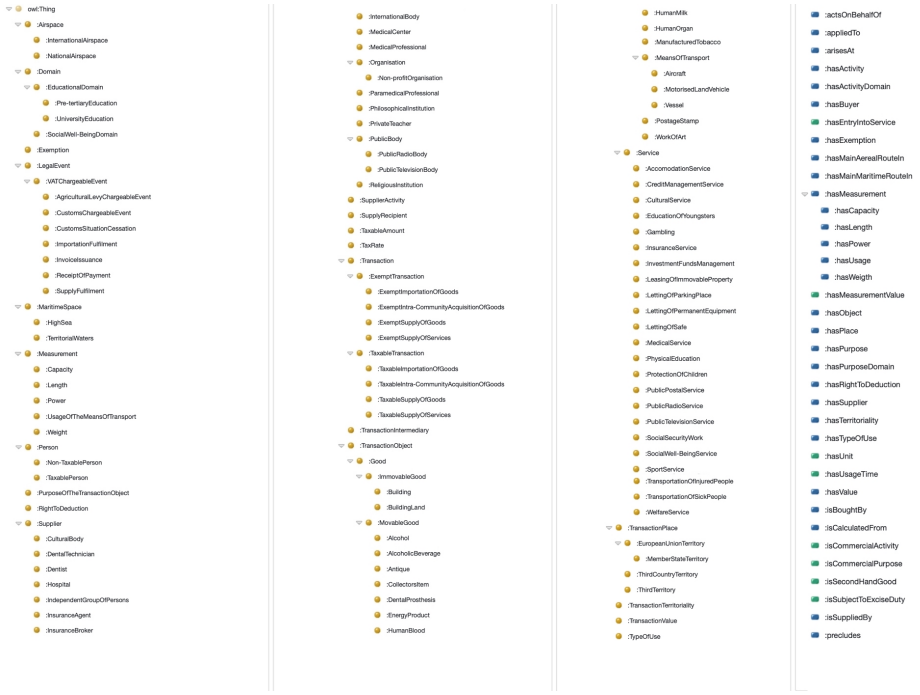


Fig. 4. All hierarchies of classes and properties.

.

To grant ontological robustness across the conceptual framework, most classes in OntoVAT are designed to be disjointed. The only class we decided not to disjoin are **VAT Chargeable Event**, **Domain**, and **Supplier**.

As can be seen in Figure 5, we did not disjoin the subclasses of "VAT Chargeable Event" to allow an instance of VAT chargeable event to belong to multiple types of chargeable event. Regarding the "Domain" concept, we preferred to allow an instance of domain to belong to multiple classes because the supplier's activity might sometimes involve an overlap of multiple domains, and because a domain might sometimes be defined as an intersection of multiple sub-domains. For the same reason, we also wanted to allow potential overlaps in the subclasses of the concept "Supplier".

These choices of allowing the overlap in the above mentioned cases (i.e., VAT chargeable events, domains and suppliers) might be made clearer with an example: an individual of the class "Dentist" could also be, in principle, an individual of the class "Private Teacher". Similarly, we decided that it was safer to leave potential overlapping among the sub-classes of "VAT chargeable event" as well as among the sub-classes of "Domain".

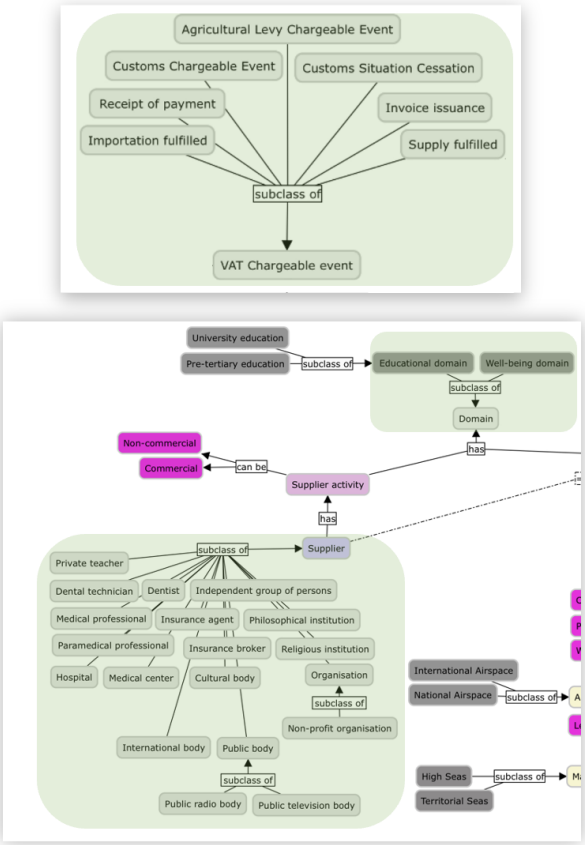


Fig. 5. The portions of OntoVAT which allow internal overlaps (i.e. where individuals can belong to multiple classes) are the subclasses of "VAT Chargeable Event" (image in top), "Domain" and "Supplier" (both depicted in the image at the bottom).

.

4.4 Alignment with upper ontologies

To make OntoVAT more robust and interoperable, we are exploring alignments to other well-known legal upper ontologies, in particular LKIF (Legal Knowledge Interchange Format) [6]. We also align to the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) and to the DOLCE+DnS Ultralite (DUL) ontology [1].

We list the alignment of our classes in Table 2, while Figure 6 shows the alignments and commitment to the upper ontologies as described in Table 2.

OntoVAT class	Aligned with class	In	Comment
Airspace	place:Place	LKIF	
Domain	expression:Qualification	LKIF	
Exemption	norm:Norm	LKIF	An “exemption” is the result of interactions between norms, which are meant to assess if an exemption occurs.
Legal Event	top:Spatio Temporal Occurrence	LKIF	
Maritime Space	place:Place	LKIF	
Measurement	dul:Unit Of Measure	DUL	
Purpose Of The Transaction Object	expression:Qualification	LKIF	
Person	legal-action:Legal Person	LKIF	
Right To Deduction	norm:Right	LKIF	
Supplier	legal-role:Legal Role	LKIF	
Supplier Activity	expression:Qualification	LKIF	
Supply Recipient	legal-role:Legal Role	LKIF	
Taxable Amount	dul:Amount	DUL	
Tax Rate	dul:Amount	DUL	
Transaction	action:Trade	LKIF	action:Trade seems more appropriate than action:Transaction
Transaction Intermediary	legal-role:Legal Role	LKIF	
Transaction Object	dolce:Substantial	DOLCE	Regarding its subclasses, “Good” aligns to “dolce:Agentive physical Object”; “Service” aligns to “dolce:SocialObject”
Transaction Place	place:Place	LKIF	
Transaction Territoriality	norm:Norm	LKIF	The concept of “territoriality” is the result of interactions between norms, which are meant to assess a given geographical space (i.e. the “Transaction Place”).
Transaction Value	dul:Amount	DUL	
TypeOfUse	expression:Qualification	LKIF	

Table 2. Alignment and interoperability with upper ontologies.

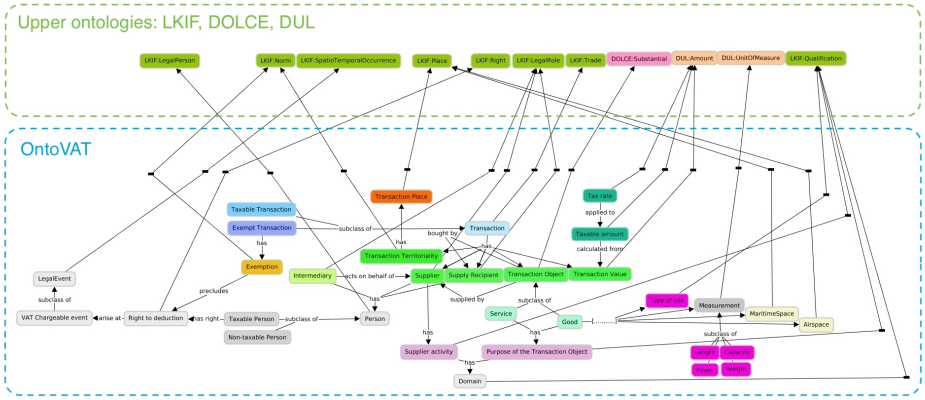


Fig. 6. OntoVAT alignments with upper ontologies.

.

5 Conclusion

In this work, we presented the first version of OntoVAT, the first formal ontology in the legal domain of VAT. The ontology has been created in cooperation with domain experts and computer scientists, and is designed to capture key VAT-related concepts in judicial decisions. The ontology is designed in OWL and is enriched with a SKOS lexicalisation in three different languages (English, Italian, and Bulgarian).

Regarding the applicative level, we are currently using this ontology to support a Natural Language Processing (NLP) pipeline, designed to extract the relevance of VAT-related concepts in our dataset of annotated legal judgments. We are also using OntoVAT to facilitate automated legal knowledge extraction from VAT-related legal documents and to build a navigation tool, through which one can find relevant judgments depending on the selected ontological concepts, through the use of semantic similarity measures.

Combining OntoVAT with an NLP pipeline is only one of the potential applications of this ontology. In the future we plan to explore other kind of targets related to legal knowledge extraction, in combination with Machine Learning.

References

1. Borgo, S., Masolo, C.: Foundational choices in dolce. In: Handbook on ontologies, pp. 361–381. Springer (2009)
2. Dimou, A., et al.: Airo: An ontology for representing ai risks based on the proposed eu ai act and iso risk management standards. In: Towards a Knowledge-Aware AI: SEMANTiCS 2022—Proceedings of the 18th International Conference on Semantic Systems, 13-15 September 2022, Vienna, Austria. vol. 55, p. 51. IOS Press (2022)
3. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with dolce. In: Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web: 13th International Conference, EKAW 2002 Sigüenza, Spain, October 1–4, 2002 Proceedings 13. pp. 166–181. Springer (2002)
4. Guarino, N., Welty, C.A.: An overview of ontoclean. Handbook on ontologies pp. 201–220 (2009)
5. Hitzler, P., Gangemi, A., Janowicz, K.: Ontology engineering with ontology design patterns: foundations and applications, vol. 25. IOS Press (2016)
6. Hoekstra, R., Breuker, J., Di Bello, M., Boer, A., et al.: The lkif core ontology of basic legal concepts. LOAIT **321**, 43–63 (2007)
7. Kerremans, K., Temmerman, R., Tummers, J.: Representing multilingual and culture-specific knowledge in a vat regulatory ontology: Support from the termontography method. In: On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops: OTM Confederated International Workshops, HCI-SWWA, IPW, JTRES, WORM, WMS, and WRSM 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings. pp. 662–674. Springer (2003)
8. Liga, D., Amitrano, D., Markovich, R.: Patronto, an ontology for patents and trademarks. In: New Frontiers in Artificial Intelligence: JSAI-isAI 2023 Workshops, AI-Biz, EmSemi, SCIDOCA, JURISIN 2023 Workshops, Hybrid Event, June 5–6, 2023, Revised Selected Papers. Springer (2024)

9. Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L.: Pronto: Privacy ontology for legal compliance. In: Proc. 18th Eur. Conf. Digital Government (ECDG). pp. 142–151 (2018)
10. Rodríguez-Doncel, V., Palmirani, M., Araszkiewicz, M., Casanovas, P., Pagallo, U., Sartor, G.: Introduction: A hybrid regulatory framework and technical architecture for a human-centered and explainable ai. In: AI Approaches to the Complexity of Legal Systems XI-XII, pp. 1–11. Springer (2020)
11. Sartor, G., Casanovas, P., Biasiotti, M., Fernández-Barrera, M.: Approaches to legal ontologies: Theories, domains, methodologies. law. Governance and Technology series. Springer (2011)
12. Stellato, A., Fiorelli, M., Turbati, A., Lorenzetti, T., Van Gemert, W., Dechandon, D., Laaboudi-Spoiden, C., Gerencsér, A., Waniart, A., Costetchi, E., et al.: Vocbench 3: A collaborative semantic web editor for ontologies, thesauri and lexicons. Semantic Web **11**(5), 855–881 (2020)
13. Temmerman, R., Kerremans, K.: Termontography: Ontology building and the sociocognitive approach to terminology description. Proceedings of CIL17 **7**, 1 (2003)

Using WikiData for Handling Legal Rule Exceptions: Proof of Concept

Wachara Fungwacharakorn[1[0000−0001−9294−3118]], Hideaki Takeda[1[0000−0002−2909−7163]], and Ken Satoh[1[0000−0002−9309−4602]]

National Institute of Informatics, Sokendai University, Tokyo, Japan
{wacharaf,takeda,ksatoh}@nii.ac.jp

Abstract. Since social expectations and circumstances always change, it is challenging to refine formalized legal rules for handling new exceptions driven from new exceptional cases. Recent research explores using linked open datasets for handling new exceptions. However, most research uses domain-dependent datasets developed by specific authorities. To handle legal rule exceptions across various domains and to keep knowledge updated according to social changes, we explore in this paper a framework for handling legal rule exceptions using WIKIDATA, which is a linked open dataset retrieved from various Wikimedia projects with collaborative edits. By exploring factor hierarchies extracted from WIKIDATA, the framework assists a user to find factors well describing new exceptional cases. We found that it is feasible to extract factor hierarchies if rule factors subsume case factors. However, some rule factors may not subsume case factors in practice and we suggest using such rule factors for assisting users in refining rules, cases, or knowledge bases so that all rule factors subsume case factors. We demonstrate the framework using two cases from the European Court of Human Rights and discuss possible improvements for this framework using current technologies, such as WIKIDATA-Lite, Word2vec, and ChatGPT.

Keywords: Legal reasoning · Linked Open Data · Wikidata

1 Introduction

Researchers have been long interested in representing legal rules using various formalizations, including CATALA [11], Defeasible Logic [16], PROLEG [20], and PROLOG [21–23]. In those formalizations, representations of legal rules are commonly divided into conclusions, requisites (positive conditions of the rules), and exceptions (negative conditions of the rules). One of the most challenging aspects in formalizing legal rules is to handle new exceptions, which are frequently updated according to changing circumstances and social expectations. Thus, legal reasoning is viewed as moving classification systems [10] as judges may introduce new exceptions to distinguish cases while judging them [5]. The needs of handling new exceptions also occur in Artificial Intelligence (AI) as it is infeasible to express all of the exceptions in the first place [24].

In recent years, there has been a growing interest in using linked open datasets to aid in handling exceptions to legal rules, including using Privacy Ontology (PrOnto) [15] for handling exceptions in General Data Protection Regulation (GDPR) [18] and using ASAM OpenXOntology [3] for handling exceptions to traffic rules [26]. However, those linked datasets are domain-dependent hence it uses exhaustive consideration with specific objectives or authorities to build them.

In contrast to most previous works, we investigate a framework using Wikidata [25] for handling legal rule exceptions. Wikidata is a domain-independent and collaboratively-created linked open dataset supported by Wikimedia Foundation. The dataset is retrieved from several Wikimedia projects, and open for both humans and machines to read and edit it. It shows that Wikidata can be used for analogical reasoning but with manual effort [7]. We expect that using Wikidata may open a possibility for handling legal rule exceptions across various domains and update according to circumstances and social expectations.

In our framework, we assume that requisites of legal rules can be extracted into sets of factors, which are associated with items in Wikidata. As we assume that all factors of rules subsume factors of cases, the framework navigates the user to explore other items in Wikidata, assisting the user to invent new factors well describing exceptions to the legal rules. Nonetheless, factors of rules do not always subsume factors of cases in practice. Hence, the framework needs to assist users in exploring possible factor hierarchies and refining rules, cases or knowledge bases so that all factors of rules subsume some factors of cases. To demonstrate the framework, we use two cases from the European Court of Human Rights (ECHR), *Eweida v. UK (2013)* and *S.A.S. v France (2014)*, involving the rights to freedom of religion (Article 9 of the European Convention on Human Rights).

This paper is structured as follows. Section 2 gives backgrounds on example cases and Wikidata. Section 3 presents a framework using Wikidata for handling legal rule exceptions. Section 4 discusses possible improvements for the framework. Finally, section 5 provides a conclusion of this paper.

2 Preliminaries

2.1 Example Cases

In this paper, we demonstrate handling legal rules and exceptions using two cases related to Article 9 of the European Convention on Human Rights, which states as follows.

Article 9 – Freedom of thought, conscience and religion

1. Everyone has the right to freedom of thought, conscience and religion; this right includes freedom to change her/his religion or belief and freedom, either alone or in community with others and in public or private, to manifest her/his religion or belief, in worship, teaching, practice and observance.

2. Freedom to manifest one's religion or beliefs shall be subject only to such limitations as are prescribed by law and are necessary in a democratic society in the interests of public safety, for the protection of public order, health or morals, or for the protection of the rights and freedoms of others.

Two cases discussed in this paper are as follows.

Eweida v. UK (2013) In 2006, Eweida, an employee of British Airways, wore a necklace with a small cross, against the uniform policy of wearing religious jewellery out of sight. The British courts ruled in favor of British Airways. The European Court of Human Rights (ECHR) found that the British government had failed to protect the rights to freedom of religion, in breach of Article 9.

S.A.S. v France (2014) In 2011, banning face covering in public places became effective in France. A Muslim French woman filed a complaint against the French state as the law prevented her from wearing the niqab (a religious face covering, leaving the eyes uncovered) in public places. The European Court ruled in favor of France as an exception to Article 9.

2.2 Wikidata

WIKIDATA [25] is a domain-independent and collaboratively-created linked open dataset supported by Wikimedia Foundation. The dataset is retrieved from several Wikimedia projects including Wikipedia, Wiktionary, Wikisource, etc. As of March 2023, the dataset contains around 102 million items covering knowledge from various domains.

In WIKIDATA, each item is identified with a prefix `wd:` and a unique identification code to distinguish two items with the same label, for instance Apple (`wd:Q312`) refers to a technology company while Apple (`wd:Q26944932`) refers to a family name. Direct relations between items are identified with a prefix `wdt:`, including general relations like a subclass of (`wdt:P279`) and specific relations like a father of (`wdt:P22`). WIKIDATA provides a query service based on SPARQL. Below shows an example query for finding all items which are direct subclasses of religious objects (`wd:Q21029893`). The query after `SERVICE` is for retrieving item labels in English.

```
SELECT ?item ?itemLabel
WHERE {
   ?item wdt:P279 wd:Q21029893.
   # direct subClassOf religious object
   SERVICE wikibase:label {
        bd:serviceParam  wikibase:language "en".}
}
```

Furthermore, WIKIDATA query service allows a query to contain a property path, which is useful for querying multiple and/or recursive relations. Below

shows an example query for finding all items which are descendent subclasses (i.e. subclasses, subclasses of subclasses, ...) of religious objects using an asterisk (*).

```
SELECT ?item ?itemLabel
WHERE {
   ?item wdt:P279* wd:Q21029893.
   # descendent subClassOf religious object
   SERVICE wikibase:label {
        bd:serviceParam wikibase:language "en".}
}
```

3 Legal Rule Formalization and Exception Handling

In this section, we account for legal rule formalizations. The formalizations tend to simulate *legal theories* – which judges or legal scholars develop from written rules in order to make consistent interpretations across courts. For example, judges in the Japanese Legal Training Institute developed a legal theory known as the Japanese Presupposed Ultimate Facts Theory or *Yoken-jijitsu-ron* [8]. Legal theories are used for formalizing legal rules into conclusions, requisites, and exceptions. When a case presents to the court, judges need to prove whether the conditions in the legal theories can subsume the facts of the case. The conclusion is derived if all requisites are provable and all exceptions are not provable. Then, if the derivation of the conclusion is not intended, the judge may introduce a new exception that is provable at least in the present case.

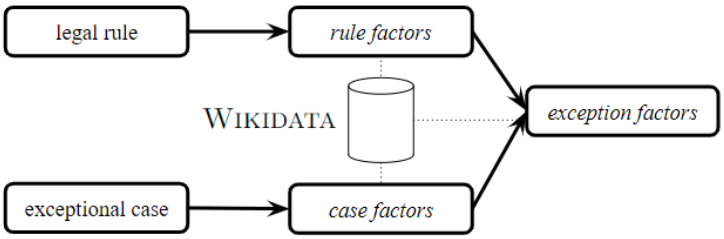


Fig. 1. Framework Overview

To formalize the process, we present our framework as depicted in Fig. 1. We assume that requisites of legal rules can be extracted into sets of factors (called *rule factors*), each of which is associated with an item in WIKIDATA. Meanwhile, cases are also assumed to be extracted into sets of factors (called *case factors*) as in several classic case-based legal reasoning systems [1, 4, 17]. Each case factor is also associated with an item in WIKIDATA. Aligning with a previous work of using background knowledge as factor hierarchies [2], we assume that rule factors

subsume case factors, i.e. some items associated with case factors are descendent subclasses of items associated with rule factors.

The framework aims for handling legal rule exceptions when we find an exceptional case, of which an outcome is expected to be opposite from what the rule gives and hence an exception to the rule is needed. The ultimate goal of the framework is to find *exception factors* with the following properties, along with research on handling exceptions in formalized legal rules [6] as well as in non-monotonic reasoning [19].

1. All exception factors subsume case factors (so the exceptions are provable at least in the present case).
2. An exceptional factor does not subsume any rule factors (otherwise exception factors would override the rule).

To find exception factors that are suitable for describing exceptional situations, the framework navigates a user to explore sibling items, which are siblings of the items in descendant lists from subsuming rule factors to subsumed case factors, so that the user can find or invent such exception factors. Let demonstrate the framework with the example cases *S.A.S. v France (2014)*. Suppose there is a legal rule formalized in PROLOG, explaining the rights to wear religious objects on somebody as follows.

```
rights_to_wear(X) :- religious_objects(X).
```

where `religious_objects` is associated with an item `religious objects` (`wd:Q21029893`) in WIKIDATA. Meanwhile, a factor in the case is associated with an item `niqab` (`wd:Q210583`) in WIKIDATA. Fortunately, the item `niqab` is a descendant subclass to the item `religious objects` inWIKIDATA. By using subclass queries as below, the framework can cooperate with users to extract a descendant list from `religious_objects` (`wd:Q21029893`) to `niqab` (`wd:Q210583`) as in Fig. 2.

```
SELECT ?item ?itemLabel
WHERE {
   ?item wdt:P279* wd:Q21029893.
   wd:Q210583 wdt:P279* ?item.

   SERVICE wikibase:label {
        bd:serviceParam wikibase:language "en".}
}
```

Then, the framework navigates the user through the hierarchy to generalize exceptional situations from the cases. For example, since `Jilbab` (`wd:Q1248904`) is a subclass of `hijab` (`wd:Q109326154`) but it allows to open the whole face, it leads to the question whether a jilbab is included in an exception, i.e. shall we allow to ban an open-faced jilbab, in the same manner of banning a niqab, which is a closed face covering. Suppose a jilbab is excluded from the exceptional situation, i.e. we shall not ban a jilbab. Hence, we may invent a new item *close-face*

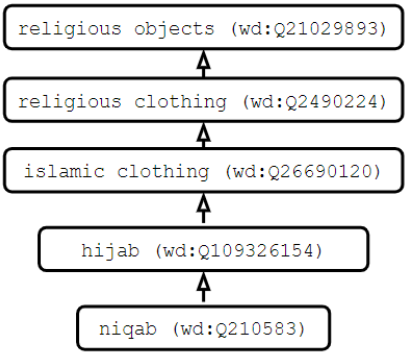


Fig. 2. A descendant list extracted from from `religous_objects` (`wd:Q21029893`) to `niqab` (`wd:Q210583`)

covering that is suitable for handling the exception. The new item may be introduced into WIKIDATA, includes `niqab` (`wd:Q210583`) as its subclass but excludes `Jilbab` (`wd:Q1248904`) from its subclass. The new item is then associated with a new predicate `closeface_covering` in revised PROLOG legal rules as follows.

```
rights_to_wear(X) :- religous_objects(X), not exception(X).
exception(X) :- closeface_covering(X).
```

However, it is impossible to extract a descendant list if an item is not a descendant subclass to another item in WIKIDATA. Let demonstrate the problem with the example case *Eweida v. UK (2013)*. We have that the case is associated with an item `cross necklace` (`wd:Q5188464`) in WIKIDATA. Unfortunately, `cross necklace` (`wd:Q5188464`) is not a descendant subclass to `religious_objects` (`wd:Q21029893`).

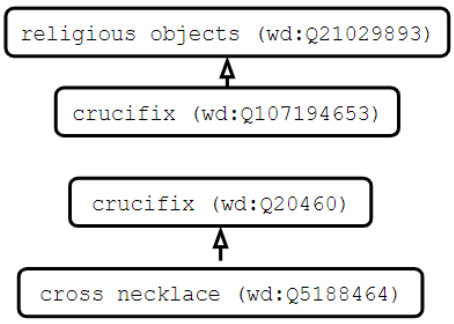


Fig. 3. Two disconnected descendant lists related to `religious_objects` (`wd:Q21029893`) and `cross necklace` (`wd:Q5188464`)

Fig. 3 shows two disconnected descendant lists, one with `religious objects` (`wd:Q21029893`) and another with `cross necklace` (`wd:Q5188464`) . We have that `cross necklace` (`wd:Q5188464`) is a subclass of `crucifix` (`wd:Q20460`) described as a *cross with an image or artwork of Jesus on it* while there is another item labelled `crucifix` (`wd:Q107194653`) described as a *kind of religious object*. The user may navigate the framework to join these two descendant lists by merging two items of `crucifix` into one, or putting `cross necklace` (`wd:Q5188464`) as a subclass of `crucifix` (`wd:Q107194653`), which is then a subclass of `religious objects` (`wd:Q21029893`).

Hence, the framework consists of two parts as summarized in Algorithm 1. The first part returns all descendant lists from rule factors to case factors that can be extracted from WIKIDATA. However, if some rule factors do not subsume any case factors, the second part additionally returns all items that are descendent subclasses of such rule factors. We expect that the user may revise rules, cases, or the knowledge base by using the result of the second part so that all rule factors subsume some case factors. Then, the user invokes the algorithm to return descendant lists by the first part again.

Algorithm 1 Extract WIKIDATA item sets for handling exceptions

Input a WIKIDATA item set R associated with *rule factors*, and a WIKIDATA item set C associated with *case factors*
Output a set *output* of WIKIDATA item sets

$Output = \varnothing$
for all $r \in R$ and $c \in C$ **do**
  Query for sets D of descendant lists from r to c
  if $D \neq \varnothing$ **then**
    Mark r in R
    $Output = Output \cup D$
for all unmarked $r \in R$ **do**
  Query for a set D_r of WIKIDATA items that are descendent subclasses of r
  $Output = Output \cup D_r$
return $Output$

To analyze the complexity of the algorithm, we can consider extracting a descendant list from one item to another item using breadth-first search. If an item is a descendant subclass to another item with a depth of d and each item has direct subclasses with an average size of n, it takes $O(n^{d+1})$ to extract a descendant list, and the size of the descendant list is $O(d)$, proportional to the depth. However, if an item has no target item as a descendant subclass, it takes $O(n^{D+1})$ where D is the longest depth of descendent subclasses of the item. Furthermore, since the framework extracts all descendent subclasses of the item, the size of the result is also $O(n^{D+1})$. Hence, the execution time and the size

of results are generally larger if an item associated with a rule factor does not subsume any case factors.

4 Discussion

In this paper, we present a framework for handling legal rule exceptions based on factor hierarchies extracted from WIKIDATA from factors of rules and factors of cases. We found that if all factors of rules subsume factors of cases, it is feasible to extract descendant lists. However, some factors of rules do not subsume any factors of cases in practice. To cope with this problem, the framework returns all descendant subclasses of such factors of rules to assist users in revising the rule, the case, or the knowledge base. This takes long execution time and infeasibly large results. To improve this, we consider datasets pruned from WIKIDATA, such as WIKIDATA-lite [13], as well as using word embeddings, such as Word2vec [12], in searching and pruning the result.

Beside exploring items in the descendant list, we suggest using the framework to navigate users through sibling items, i.e. items that are siblings of the items in the descendant list. Since sibling items do not occur in the case presented to the court, it may be suitable for legislation rather than jurisdiction to discuss several concepts outside the case. Navigating through sibling items may assist users to exhaustively consider exceptional situations of the case; however, which terms should be used for describing exceptional situations is still handcrafted. The terms well describing exceptional situations seldom exist in WIKIDATA as they probably realized as exceptions of the rule if the terms already exist. Hence, mismatches between concepts in rules and in knowledge bases would lead to revisions of the rules and the knowledge bases, aligned with previous research [9].

As currently, ChatGPT [14], an artificial intelligence chatbot based on GPT3, has become popular, we tested whether ChatGPT can explain exceptions of *S.A.S. v. France (2014)*. We ask ChatGPT "What is an exceptional situation of *S.A.S. v. France (2014)* ?" in a new session and here is a part of the answer.

> ... One exceptional situation of this case was the argument put forth by the French government that the ban on full-face veils was necessary for public safety and security reasons. ...

We can see that ChatGPT can identify the exceptional situation correctly. However, it is still a question of whether ChatGPT can consider exceptional situations correctly for a case that has not been decided yet. Besides that, we test whether ChatGPT can invent a term that can describe exceptional cases but excludes non-exceptional cases with the following *filling in the blank* question in a new session and here is an answer.

> Q: Fill in the blank:
> input: brothers and sisters, but not fathers
> output: siblings

input: burqa and niqab, but not jilbab
output: ___
A: face veils

The answer is profound but not precise (*full-face veils* is a precise answer). However, the performance of ChatGPT in the task of describing exceptional situations is an interesting future work.

5 Conclusion

In this paper, we explore a framework for handling legal rule exceptions using WIKIDATA, a collaboratively-edited linked open dataset supported by Wikimedia foundation. The inputs of the framework are two sets of *rule factors* and *case factors* representing legal rules and cases respectively. We assume that each factor is associated with an item in WIKIDATA, and all rule factors subsume case factors. To find factors that are suitable for describing exceptional situations, the framework navigates a user to explore factor hierarchies extracted from WIKIDATA, beginning with rule factors to subsumed case factors, so that the user can find or invent such exception factors. However, it is possible that some rule factors do not subsume any case factors in practice so the framework navigates the user through WIKIDATA items that can be subsumed by such rule factors. In the future, we are interested in using several technologies, such as WIKIDATA-Lite, Word2Vec, ChatGPT, for pruning unnecessary navigation as well as creatively inventing new terms for describing exceptional situations.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Numbers, JP17H06103 and JP19H05470 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4.

References

1. Aleven, V.: Teaching case-based argumentation through a model and examples. Ph.D. thesis, University of Pittsburgh (1997)
2. Aleven, V.: Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment. Artificial Intelligence **150**(1-2), 183–237 (2003)
3. ASAM: Asam openxontology. https://www.asam.net/project-detail/asam-openxontology/, accessed: 2023-03-06
4. Ashley, K.D.: Modeling Legal Arguments: Reasoning with Cases and Hypotheticals. The MIT Press, Cambridge (Massachusetts) (1990)
5. Ashley, K.D.: Artificial intelligence and legal analytics: new tools for law practice in the digital age. Cambridge University Press, Cambridge, England (2017)
6. Fungwacharakorn, W., Satoh, K.: Generalizing culprit resolution in legal debugging with background knowledge. In: Legal Knowledge and Information Systems. Frontiers in Artificial Intelligence and Applications, vol. 334, pp. 52–62. IOS Press (2020)

7. Ilievski, F., Pujara, J., Shenoy, K.: Does wikidata support analogical reasoning? In: Knowledge Graphs and Semantic Web: 4th Iberoamerican Conference and third Indo-American Conference, KGSWC 2022, Madrid, Spain, November 21–23, 2022, Proceedings. pp. 178–191. Springer (2022)
8. Ito, S.: Basis of Ultimate Facts. Yuhikaku (2001)
9. Kurematsu, M., Tada, M., Yamaguchi, T.: A legal ontology refinement environment using a general ontology. In: Proceedings of Workshop on Basic Ontology Issues in Knowledge Sharing, International Joint Conference on Artificial Intelligence. vol. 95 (1995)
10. Levi, E.H.: An introduction to legal reasoning. University of Chicago Press, Chicago, USA (2013)
11. Merigoux, D., Chataing, N., Protzenko, J.: Catala: A Programming Language for the Law. In: International Conference on Functional Programming. pp. 1–29. Proceedings of the ACM on Programming Languages, ACM, Virtual, South Korea (Aug 2021)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
13. Nguyen, P., Takeda, H.: Wikidata-lite for knowledge extraction and exploration. arXiv preprint arXiv:2211.05416 (2022)
14. OpenAI: Chatgpt. https://chat.openai.com/chat, accessed: 2023-03-06
15. Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L.: Pronto: Privacy ontology for legal reasoning. In: Electronic Government and the Information Systems Perspective: 7th International Conference, EGOVIS 2018, Regensburg, Germany, September 3–5, 2018, Proceedings 7. pp. 139–152. Springer (2018)
16. Prakken, H.: Logical tools for modelling legal argument: a study of defeasible reasoning in law. Kluwer Academic Publishers (1997)
17. Rissland, E.L., Ashley, K.D.: A case-based system for trade secrets law. In: Proceedings of the 1st international conference on Artificial intelligence and law. pp. 60–66. Association for Computing Machinery, New York, NY, USA (1987)
18. Robaldo, L., Bartolini, C., Palmirani, M., Rossi, A., Martoni, M., Lenzini, G.: Formalizing GDPR provisions in reified I/O logic: the DAPRECO knowledge base. Journal of Logic, Language and Information **29**, 401–449 (2020)
19. Sakama, C.: Nonmonotomic inductive logic programming. In: International Conference on Logic Programming and Nonmonotonic Reasoning. pp. 62–80. Springer (2001)
20. Satoh, K., Asai, K., Kogawa, T., Kubota, M., Nakamura, M., Nishigai, Y., Shirakawa, K., Takano, C.: PROLEG: An Implementation of the Presupposed Ultimate Fact Theory of Japanese Civil Code by PROLOG Technology. In: Onada, T., Bekki, D., McCready, E. (eds.) New Frontiers in Artificial Intelligence. pp. 153–164. Lecture Notes in Computer Science, Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
21. Satoh, K., Kubota, M., Nishigai, Y., Takano, C.: Translating the Japanese Presupposed Ultimate Fact Theory into logic programming. In: Proceedings of the 2009 Conference on Legal Knowledge and Information Systems: JURIX 2009: The Twenty-Second Annual Conference. pp. 162–171. IOS Press, Amsterdam, The Netherlands (2009)
22. Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P., Cory, H.T.: The British Nationality Act as a logic program. Communications of the ACM **29**(5), 370–386 (Apr 1986)

23. Sherman, D.M.: A Prolog model of the income tax act of Canada. In: Proceedings of the 1st international conference on Artificial intelligence and law. pp. 127–136. Association for Computing Machinery, New York, NY, USA (1987)
24. Thielscher, M.: The qualification problem: A solution to the problem of anomalous models. Artificial Intelligence **131**(1-2), 1–37 (2001)
25. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)
26. Wang, Y., Grabowski, M., Paschke, A.: An ontology-based model for handling rule exceptions in traffic scenes. In: Proceedings of the International Workshop on AI Compliance Mechanism (WAICOM 2022). pp. 87–100 (2022)

PaTrOnto, an ontology for patents and trademarks*

Davide Liga[1][0000−0003−1124−0299], Daniele Amitrano[2], and Réka Markovich[1]

[1] University of Luxembourg, Esch-sur-Alzette, Luxembourg
{davide.liga,reka.markovich}@uni.lu
[2] Trevisan & Cuonzo, Milano, Italy
damitrano@trevisancuonzo.com

Abstract. In this work, we introduce PaTrOnto, a multilingual ontology designed for legal knowledge extraction in the domain of patents and trademarks. To the best of our knowledge, this is the first attempt to build an ontology which comprehensively covers the domain of patents and trademarks in a multilingual scenario. PaTrOnto is an OWL ontology with SKOS multilingual lexicalisation, designed to capture the most most important concepts which occur within legal judgments related to patents and trademarks. We release the first version of this ontology in English, Italian and Bulgarian. The relevance of this ontology is that it allows for both reasoning (being written in OWL) and knowledge extraction (thanks to the use of some SKOS properties which provide each ontological concept with informations such as synonyms, examples, definitions, normative references). Furthermore, it has been created in close cooperation with legal experts and computer scientists.

Keywords: Legal Knowledge Representation · Ontology · Patent · Trademarks · AI&Law.

1 Introduction

In recent times, the Artificial Intelligence and Law (AI&Law) sector has undergone substantial growth, driven by advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP). This has led to the creation of numerous applications designed to support legal experts, enhance the availability of justice, and streamline legal system operations. The community has witnessed significant and noteworthy expansion during this period, propelled by the progress made in AI and NLP.

* This works has been supported by the Analytics for Decision of Legal Cases (ADELE), founded by the European Union's Justice Programme (grant agreement No. 101007420); Davide Liga was supported by the project INDIGO, which is financially supported by the NORFACE Joint Research Programme on Democratic Governance in a Turbulent Age and co-funded by AEI, AKA, DFG and FNR and the European Commission through Horizon 2020 under grant agreement No 822166

In the AI and Law community, one of the primary objectives is to identify and develop comprehensive and suitable methods for representing legal knowledge. This involves exploring various techniques and strategies to effectively capture the complexities and nuances of legal concepts, principles, and reasoning. By doing so, the community aims to enhance the accuracy and efficacy of AI-driven tools and applications designed to support legal professionals, improve access to justice, and streamline the functioning of legal systems.

Utilizing ontologies, which are structured representations of knowledge within a particular domain, can help attain these goals, as they offer a means to precisely depict intricate symbolic information in a format that machines can interpret, all while maintaining the benefits of modularity and interoperability. They can be particularly powerful also in combination with other methods of AI, both symbolic and non-symbolic.

In this work, we propose a first version of PaTrOnto (the Patent and Trademark Ontology), which is designed to facilitate both reasoning and knowledge extraction from legal judgments in the context of patent and trademarks.

We will start with Section 2, discussing some related studies. Then, we will focus on PaTrOnto in Sections 3 and 4, where we will respectively discuss about the general methodology we employed and the more specific structure of PaTrOnto. Finally, we will conclude with some ideas for the future in Section 5.

2 Related Works

Historically, ontologies have played a significant role both as domain-specific tools and as upper-ontologies. As domain-specific tools, they have been extensively employed to capture knowledge and concepts unique to particular fields, allowing for more effective organization, retrieval, and analysis of information. This has proven invaluable across various disciplines, including medicine, finance, and law, among others. In the context of upper-ontologies, they have served as foundational structures, providing a framework for integrating and connecting multiple domain-specific ontologies. This has facilitated interoperability and collaboration between different knowledge domains, promoting a more comprehensive understanding of complex, interdisciplinary problems. Consequently, ontologies have become indispensable assets in the realm of knowledge representation and management, also in the field of law [18].

In the field of AI&Law, we can find examples of both domain-specific ontologies and upper ontologies. Starting from the higher levels of abstraction, one can find upper ontologies such as the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [1], the Basic Formal Ontology (BFO) [16] or the Suggested Upper Merged Ontology (SUMO) [15]. These ontologies provide a foundational structure for integrating and aligning various domain-specific ontologies, which allows for improved interoperability and collaboration across different fields. However, one can also find domain-specific upper ontologies, i.e. ontologies which are located at abstract layers of abstraction but with the goal of representing the upper conceptual ideas of a specific domain. For example,

in the legal domain we can find ontologies such as the Legal Knowledge Interchange Format (LKIF) [8], or the Ontology of Professional Judicial Knowledge (OPJK) [3]. The focus of domain-specific upper ontologies is to capture the unique concepts and relationships within a domain, thus enabling more precise representation and analysis of domain-specific (e.g. legal) information. These two types of upper ontologies serve as a backbone for connecting legal knowledge with other disciplines, fostering a more comprehensive understanding of complex, interdisciplinary legal issues. Consequently, both domain-specific and upper ontologies have become crucial components in the advancing landscape of AI&Law.

Going towards lower layers of abstraction (i.e. towards a more domain-specific dimension), we can find ontologies designed to represent specific legal domains, such as privacy law [17] or the recent Artificial Intelligence Act ontology [4]. Our contribution is located in this level of abstraction, since we are proposing an ontology related to patents and trademarks. In this regard, there have been only few studies attempting to build ontologies in these two areas. Some study focused on specific analytical angles such as infringement [20] [12] [13] [9]. Other works developed patent ontologies focused on the specific technical or technological characteristics [11] [21].

Unlike the previous few works on patent ontologies, PaTrOnto has the broader scope of integrating patent and trademarks into the same conceptual framework, focusing in particular on the key conceptual features which judges consider when producing judgements related to patents and trademarks. The idea of creating an ontology for modeling these two areas is due to the fact that these two areas share several similar juridical concepts.

As a side note, we emphasize that PaTrOnto incorporates support for the recently introduced Unitary Patent, which is a novel type of patent available at the European level.

3 Methodology

We started building this ontology from a collection of annotated judgements. Our original idea was to create two different ontologies, one for the domain of patents and the other one for domain of trademarks. However, we realised that the most critical underlying concepts were actually shared between these two domains in a almost symmetrical way (this symmetry is even more evident when watching Figure 8, in the next Section).

Regarding the methodology, we were inspired by [17]. More precisely, we followed a top-down approach which includes the reuse of pre-existing ontology patterns [7] and which is performed on legal sources (i.e. legal judgements). Our results are strengthened by the committment to foundational and upper ontologies (in particular LKIF [8], DOLCE [5] and DUL [2]), and we followed the principles in the OntoClean [6] method, according to which each ontological concept can be evaluated based on three meta-properties:

1. “identity” (a class must be uniquely identifiable)

2. "unity" (instances of a class must form meaningful and cohesive wholes)
3. "rigidity" (referring to whether a property is essential to the instances of a class or if it can change over time)

Our validation process engaged a highly interdisciplinary team, which included lawyers, computer scientists, logicians, and philosophers. This diverse composition enabled us to incorporate a comprehensive range of expertise from various disciplines.

Our approach can be summarised as follows:

(i) a group of legal experts selected nearly 500 legal judgements related to the domain of patents and trademarks in Italian and Bulgarian;
(ii) the judgements were analyzed and the portions of text related with the judges' motivations were annotated;
(iii) Italian and Bulgarian legal experts analysed the most important concepts mentioned in the judgements, checking these concepts against their respective statutory backgrounds;
(iv) our technical team received the selected concepts and portions of text from the legal experts to map them into the ontology;
(v) for each element of the ontology our legal experts provided a range of linguistic variations/synonyms, a definition, the most common examples instantiating that concept, the most common related terms, and any relevant normative references related to the concept;
(vi) the gathered results were validated by the legal team that returned them to the technical team who implemented the new information in the ontology;
(vii) the steps from (iii) to (vi) were iterated several times to refine the ontology;

Currently, we are working on developing an algorithm that utilizes PaTrOnto to establish the relevance of an ontological concept in judgments pertaining to patents and trademarks. This process can be summarised as follows:

1. legal experts were asked to select from PaTrOnto the ontological concepts which are considered more relevant in the judges' decisions.
2. considering the concepts selected in the previous step, legal experts were asked to manually annotate nearly 70% of the judgements by including the information of whether each selected concept is relevant in each judgement by associating a binary value, where 0 means "non relevant" and 1 means "relevant" (the concept is considered relevant if the court's decision concerns that concept from the substantial point of view);
3. an algorithm designed by the technical team encodes the information contained in the ontology to predict whether or not a concept is relevant (comparing the results with the gold standard defined in the previous step);

At present, we are finalizing step 2 and executing step 3. Our preliminary results shows that by using PaTrOnto enables us to capture the most significant relevant concepts within the judges' decisions.

This approach can be adapted and utilized across various fields. For instance, we implemented the same methodology in the creation of another ontology associated with the VAT (Value-Added Tax) domain, which we called OntoVAT [14]. The primary distinction between OntoVAT and PaTrOnto pertains to the previously mentioned step (iii), as the statutory context of VAT domain differs significantly, especially in terms of harmonization at the European level. While for OntoVAT we relied on the European VAT Directive, which provides a quite harmonised framework, the analysis of the statutory background for patents and trademarks was more heterogeneous.

4 The design of PaTrOnto

PaTrOnto is a multilingual OWL ontology, featuring a SKOS lexicalization and available in English, Italian, and Bulgarian. The OWL+SKOS multilingual lexicalization addresses the challenge of semantic inconsistencies in multilingualism, as highlighted in prior research by [10]. The ontology, implemented using VocBench 3 [19], presently consists of 191 concepts (meaning OWL classes) and 107 properties (relations between classes). A detailed numerical breakdown can be found in Table 1.

Element	Quantity
Number of classes	191
Number of properties	101
Number of datatype properties	6
Number of transitive properties	0
Number of disjoint class pairs	904
Number of subclass relations	157

Table 1. PaTrOnto in numbers.

By employing SKOS, every ontological concept (that is, each OWL class) is enhanced with particular properties that are integrated into the SKOS data model, specifically:

- skos:definition
- skos:scopeNote
- skos:altLabel
- skos:hiddenLabel
- skos:example

Incorporating these properties into each ontological concept (across English, Italian, and Bulgarian) enables the inclusion of vital information within the ontology, which in turn makes PaTrOnto highly expressive and capable of representing complex legal information.

In particular, **skos:definition** contains the definition of each single OWL class (i.e., the definition of each single concept). In **skos:scopeNote**, we added relevant specifications about the skos:definition field (whenever was necessary to further specify the interpretative angle of the chosen definition). Furthermore, scopeNotes also contain all relevant normative references (if any) describing the concept. We also added any relevant synonyms in the three different languages as **skos:altLabel** properties. In **skos:example**, we added some examples of the concept (this can be considered like defining subclasses of the concept). Finaly, the property **skos:hiddenLabel** is used to store terms in natural language which might signal the presence of the concept in the text (this can be useful for any application layers built on top of PaTrOnto).

Specifically, the **skos:definition** comprises a descriptive definition of each individual ontological concept (i.e., the meaning of each distinct OWL class). Within **skos:scopeNote**, we incorporated additional details about the definition, therefore integrating the information already provided in the skos:definition field (this was done only when it was necessary to further clarify the interpretation of the ontological concept). We also employed the skos:scopeNote property to add any pertinent normative reference related to the concept, if applicable. Using **skos:altLabel**, we also included relevant synonyms in all three languages, while the **skos:example** property has been employed to describe examples for the concept. Lastly, the **skos:hiddenLabel** property stores natural language terms that may indicate the presence of the concept in the text, which can be beneficial for any application layers built on top of PaTrOnto.

To ensure a consistent and harmonious conceptual framework, we developed the PaTrOnto ontology using concepts that are applicable across multiple countries, with guidance from Italian and Bulgarian lawyers. As a result, the semantic meaning of concepts is generally harmonious between Italy and Bulgaria. This means that a single skos:definition in English is provided for each OWL class, and it is translated into Italian and Bulgarian without modifications. However, in a few instances, the definitions of concepts (i.e., their semantic meaning) differ at the national level. In such cases, national definitions take precedence, and the skos:definition in Bulgarian/Italian will not be a mere translation from English; instead, it will be a distinct definition that aligns with the respective national legislation. Furthermore, when additional clarification is needed to explain the scope of the concepts' meaning (at Bulgarian, Italian, and European levels), we employed a skos:scopeNote property in Bulgarian/Italian/English. Finally, since national legislations may use alternative terms, we treated these alternative terms as synonyms (skos:altLabel) in Italian/Bulgarian.

In summary, we address the multilingual challenge by customizing skos properties such as skos:definitions, skos:scopeNotes, and skos:altLabels when necessary, without compromising the consistency of the ontological concepts or their relationships. Figure 1 shows an example of how multilinguality is handled for a specific concept/class.

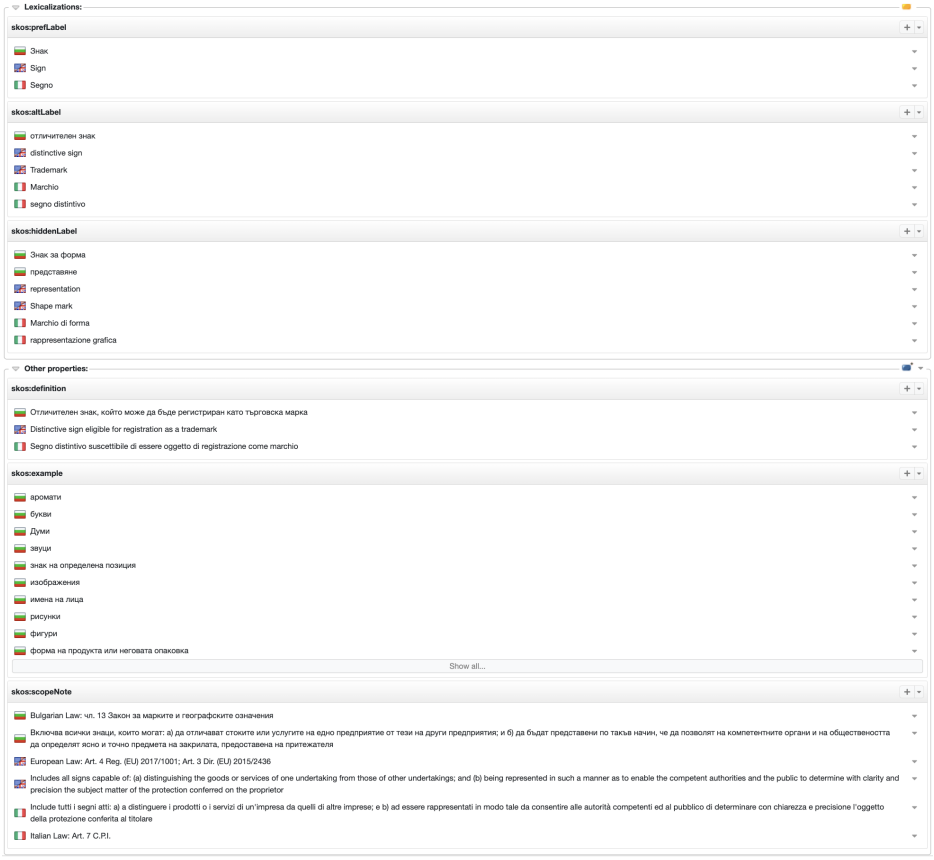


Fig. 1. An example of multilingual lexicalisation, related to the OWL class (i.e. the concept) "Sign".

We meticulously assigned a definition to each concept, prioritizing definitions derived from domain-specific legislative sources when the concept exists within that domain.

Whenever the concept is not mentioned in either national or European legislative sources, we sought a definition in the case law of the Court of Justice of the European Union (CJEU). When the concept is not defined in legislation or CJEU case law neither, as is often the case with "factual concepts", we provided a definition based on a straightforward description from legal encyclopedias or dictionaries.

By doing so, we ensured that the definition of each concept is firmly rooted in legal sources, which is essential in ensuring that the ontology can be effectively utilized in Natural Language Processing pipelines in the context of automated legal knowledge extraction.

4.1 Commitment and scope

To grant ontological robustness across the conceptual framework, most classes in PaTrOnto are designed to be disjointed. However, we decided to keep some potential overlaps in some cases.

For example, we did not disjoin all the subclasses of "Authority Of Industrial Property Right", since the same industrial property right (IPR) authority can deal with both patent and trademarks (Figure 2).

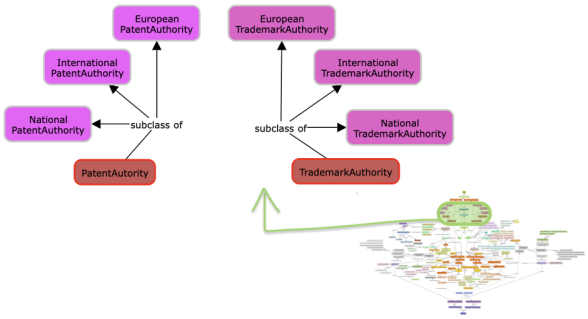


Fig. 2. Portion of PaTrOnto related to the IPR authorities.

.

We also allowed potential overlap under the class "Invention", because an instance can belong to all the subclasses (Figure 3).

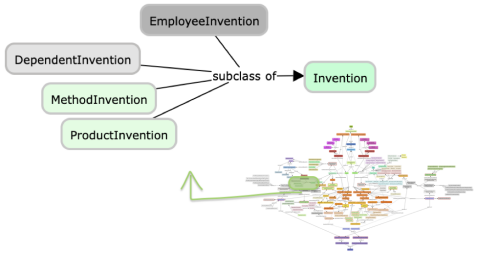


Fig. 3. Portion of PaTrOnto related to the class "Invention".

.

We also allowed overlaps under the class "Trademark", since an instance can belong to all the subclasses (Figure 4).

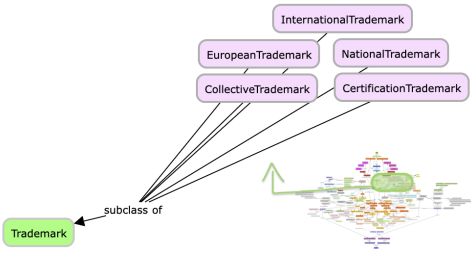


Fig. 4. Portion of PaTrOnto related to the class "Trademark".

.

Other overlaps are allowed under class "Sign" (see Figure 5), where we applied disjointness (i.e. prevented the overlap of instances) only in three cases:

- **denominative**, **figurative**, and **composite**
- **unregistered** and **registered**
- **strong** and **weak**

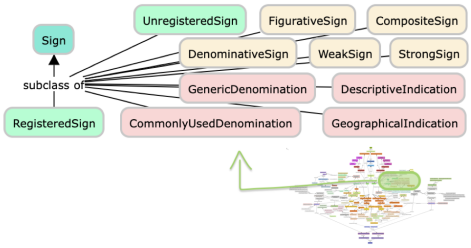


Fig. 5. Portion of PaTrOnto related to the class "Sign".

.

Regarding the invalidity, please note that "Invalidity Of Industrial Property Right" is the superclass for the "Patent Invalidity" and "Trademark Invalidity" (which are disjoint). Under "Patent Invalidity", we disjoined:

- Patent's **partial** and **total** invalidity

Under "Trademark Invalidity", we disjoined:

- Trademark's **absolute** and **relative** invalidity
- Trademark's **partial** and **total** invalidity

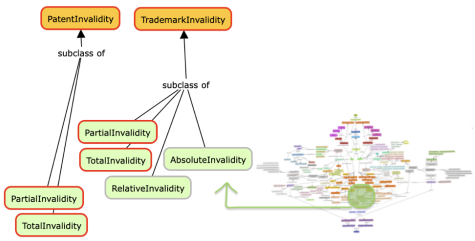


Fig. 6. Portion of PaTrOnto related to the classed "Patent Invalidity" and "Trademark Invalidity".

.

Finally, we did not disjoin the subclasses of "Owner Of Industrial Property Right" (i.e. **"Patent Owner"** and **"Trademark Owner"**) simply because an individual of the "Patent Owner" class can also be an individual of the "Trademark Owner" class. Similarly, we allowed overlapping between three classes referred to dates ("Registration Date Of Industrial Property Right", "Priority Date", and "Application Filing Date Of Industrial Property Right"). In this last case, we applied disjointness only in the following three cases:

- **"Patent Application Filing Date"** disjoined with **"Patent Granting Date"**
- **"Trademark Application Filing Date"** disjoined with **"Trademark Registration Date"**
- **"Priority Date"** disjoined with **"Patent Granting Date"**

4.2 PaTrOnto's language-specific concepts

In PaTrOnto we decided to add a class which is not present to the Bulgarian law, since it is very often taken into account within the reasoning of judges of other non-Bulgarian judgements. This class is the "Problem-Solution Approach", which is a way to evaluate the "Inventive Step" of a "Patentable Solution". In Figure 7, we put in evidence the relative concepts and relationship between concepts.

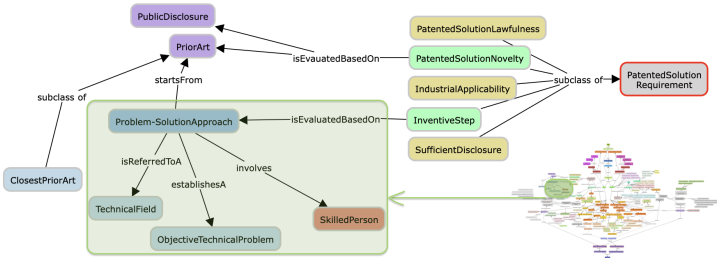


Fig. 7. Classes which do not apply to the Bulgarian system (see green area).

.

4.3 Alignment with upper ontologies

To enhance the robustness and interoperability of PaTrOnto, we are investigating potential alignments with other prominent legal upper ontologies, specifically LKIF (Legal Knowledge Interchange Format) [8]. We list the current alignments of our classes in Table 2, while Figure 8 depicts a simplified conceptual map which provides a clearer understanding of PaTrOnto, showing most of its classes and properties [3].

PaTrOnto class	Aligned with class	In
Agreement	Legal Document	LKIF
Application Filing Date of Industrial Property Right	Spatio Temporal Occurrence	LKIF
Application of Industrial Property Right	Legal Document	LKIF
Authority of Industrial Property Right	Agent	LKIF
Duration of Industrial Property Right	Temporal Occurrence	LKIF
Effectiveness of Industrial Property Right	Norm	LKIF
Exclusive Patrimonial Right to Industrial Property	Right	LKIF
Industrial Property Requirement	Norm	LKIF
Industrial Property Use	Action	LKIF
Infringement of Industrial Property Right	Action	LKIF
Intellectual Property	Creation	LKIF
Intellectual Property Right	Right	LKIF
Invalidity of Industrial Property Right	Norm	LKIF
Inventor	Agent	LKIF
Lapse Cause of Industrial Property Right	Spatio Temporal Occurrence	LKIF
Licence Duration of Industrial Property Right	Temporal Occurrence	LKIF
Moral Right to Patented Solution	Right	LKIF
Objective Technical Problem	Mental Entity	LKIF
Owner of Industrial Property Right	Legal Role	LKIF
Patent Limitation	Norm	LKIF
Patent Part	Owl:Thing	/
Patent Text	Owl:Thing	/
Patent Text Translation	Owl:Thing	/
Principle of Exhaustion	Norm	LKIF
Prior Art	Observation	LKIF
Prior Use	Action	LKIF
Priority Date	Spatio Temporal Occurrence	LKIF
Problem-Solution Approach	Process	LKIF
Public Disclosure	Action	LKIF
Registration Date of Industrial Property Right	Spatio Temporal Occurrence	LKIF
Registration Number of Industrial Property Right	Owl:Thing	/
Reputation	Observation	LKIF
Scope of Patent Protection	Norm	LKIF
Secondary Meaning	Observation	LKIF
Sign	Owl:Thing	/
Skilled Person *(note: fictional agent)*	Agent	LKIF
Technical Field	Qualification	LKIF
Territoriality of Industrial Property Right	Place	LKIF
Trademark Class	Qualification	LKIF
Trademark Validation	Observation	LKIF
Triple Identity Test	Process	LKIF
Validity of Industrial Property Right	Norm	LKIF

Table 2. Alignment and interoperability with upper ontologies.

[3] In this simplified map, relations such as "has" connecting to a target concept are represented in OWL as "hasTargetConcept", while relations such as "can be" are translated in OWL as datatype properties with a boolean value.

Fig. 8. Simplified map of the main concepts and relations in PaTrOnto.

5 Conclusion

In this study, we introduced the initial version of PaTrOnto, the first formal ontology in the legal domain of patents and trademarks. Developed in collaboration with domain experts and computer scientists, the ontology is designed to encapsulate critical domain-specific concepts found in legal judgments. PaTrOnto is structured in OWL and enriched with a SKOS lexicalization in English, Italian, and Bulgarian.

As for its application, we are currently employing PaTrOnto in a scenario where it supports a Natural Language Processing (NLP) pipeline for extracting the relevance of domain-specific concepts from our dataset of annotated legal judgments. This combination of PaTrOnto and an NLP pipeline represents just one of the potential uses for this ontology. In the future, we plan to investigate other objectives related to legal knowledge extraction, including the incorporation of Machine Learning algorithms.

In general, PaTrOnto can facilitate various types of targets. Presently, we are utilizing it to enable automated legal knowledge extraction from domain-specific legal documents and to develop a navigation tool which allows users to find relevant judgments from our dataset, based on selected ontological concepts, by using semantic similarity measures.

References

1. Borgo, S., Ferrario, R., Gangemi, A., Guarino, N., Masolo, C., Porello, D., Sanfilippo, E.M., Vieu, L.: Dolce: A descriptive ontology for linguistic and cognitive engineering. Applied ontology **17**(1), 45–69 (2022)
2. Borgo, S., Masolo, C.: Foundational choices in dolce. In: Handbook on ontologies, pp. 361–381. Springer (2009)
3. Casellas, N., Casellas, N.: Modelling judicial professional knowledge: A case study. Legal Ontology Engineering: Methodologies, Modelling Trends, and the Ontology of Professional Judicial Knowledge pp. 171–240 (2011)
4. Dimou, A., et al.: Airo: An ontology for representing ai risks based on the proposed eu ai act and iso risk management standards. In: Towards a Knowledge-Aware AI: SEMANTiCS 2022—Proceedings of the 18th International Conference on Semantic Systems, 13-15 September 2022, Vienna, Austria. vol. 55, p. 51. IOS Press (2022)
5. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening ontologies with dolce. In: Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web: 13th International Conference, EKAW 2002 Sigüenza, Spain, October 1–4, 2002 Proceedings 13. pp. 166–181. Springer (2002)
6. Guarino, N., Welty, C.A.: An overview of ontoclean. Handbook on ontologies pp. 201–220 (2009)
7. Hitzler, P., Gangemi, A., Janowicz, K.: Ontology engineering with ontology design patterns: foundations and applications, vol. 25. IOS Press (2016)
8. Hoekstra, R., Breuker, J., Di Bello, M., Boer, A., et al.: The lkif core ontology of basic legal concepts. LOAIT **321**, 43–63 (2007)
9. Jiang, P., Atherton, M., Harrison, D., Malizia, A., et al.: Framework of mechanical design knowledge representations for avoiding patent infringement. In: DS 87-6

Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 6: Design Information and Knowledge, Vancouver, Canada, 21-25.08. 2017. pp. 081–090 (2017)
10. Kerremans, K., Temmerman, R., Tummers, J.: Representing multilingual and culture-specific knowledge in a vat regulatory ontology: Support from the termontography method. In: On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops: OTM Confederated International Workshops, HCI-SWWA, IPW, JTRES, WORM, WMS, and WRSM 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings. pp. 662–674. Springer (2003)
11. Lee, C.S., Wang, M.H., Hsiao, Y.C., Tsai, B.H.: Ontology-based gfml agent for patent technology requirement evaluation and recommendation. Soft Computing **23**, 537–556 (2019)
12. Li, A., Trappey, A., Trappey, C.: Intelligent identification of trademark case precedents using semantic ontology. In: Transdisciplinary Engineering for Complex Socio-technical Systems–Real-life Applications, pp. 534–543. IOS Press (2020)
13. Li, G.K.J., Trappey, C.V., Trappey, A.J., Li, A.A.: Ontology-based knowledge representation and semantic topic modeling for intelligent trademark legal precedent research. World Patent Information **68**, 102098 (2022)
14. Liga, D., Fidelangeli, A., Markovich, R.: Ontovat, an ontology for knowledge extraction in vat-related judgments. In: New Frontiers in Artificial Intelligence: JSAI-isAI 2023 Workshops, AI-Biz, EmSemi, SCIDOCA, JURISIN 2023 Workshops, Hybrid Event, June 5–6, 2023, Revised Selected Papers. Springer (2024)
15. Niles, I., Pease, A.: Towards a standard upper ontology. In: Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001. pp. 2–9 (2001)
16. Otte, J.N., Beverley, J., Ruttenberg, A.: Bfo: Basic formal ontology. Applied ontology (Preprint), 1–27 (2022)
17. Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L.: Pronto: Privacy ontology for legal compliance. In: Proc. 18th Eur. Conf. Digital Government (ECDG). pp. 142–151 (2018)
18. Sartor, G., Casanovas, P., Biasiotti, M., Fernández-Barrera, M.: Approaches to legal ontologies: Theories, domains, methodologies. law. Governance and Technology series. Springer (2011)
19. Stellato, A., Fiorelli, M., Turbati, A., Lorenzetti, T., Van Gemert, W., Dechandon, D., Laaboudi-Spoiden, C., Gerencsér, A., Waniart, A., Costetchi, E., et al.: Vocbench 3: A collaborative semantic web editor for ontologies, thesauri and lexicons. Semantic Web **11**(5), 855–881 (2020)
20. Trappey, C.V., Chang, A.C., Trappey, A.J.: Building an internet-based knowledge ontology for trademark protection. Journal of Global Information Management (JGIM) **29**(1), 123–144 (2021)
21. Zhai, D., Zhai, L., Li, M., He, X., Xu, S., Wang, F.: Patent representation learning with a novel design of patent ontology: Case study on pem patents. Technological Forecasting and Social Change **183**, 121912 (2022)

Modeling Medical Data Access with Prova

Theodoros Mitsikas[1,2], Ralph Schäfermeier[3], and Adrian Paschke[1,4]

[1] Institut für Angewandte Informatik, Leipzig, Germany
[2] National Technical University of Athens, Zografou, Greece
mitsikas@central.ntua.gr
[3] Leipzig University, Leipzig, Germany
ralph.schafermeier@gmail.com
[4] Freie Universität Berlin and Fraunhofer FOKUS, Berlin, Germany
adrian.paschke@fokus.fraunhofer.de

Abstract. We present a medical data access use case compliant to GDPR legal rules and its implementation in the rule language Prova. The use case demonstrates a typical scenario of a patient consenting to medical data sharing for specified purposes such as treatment, as well as cases where the typical rules are overridden, modifying the access rights. This requires a representation capable of expressing the interaction between parties, and state transitions caused by this interaction. We discuss the Prova implementation which utilizes non-monotonic state transitions and reactive messaging to model the interaction between the parties, which are represented as agents.

Keywords: GDPR · Knowledge Representation· Prova · Legal Reasoning

1 Introduction

Legislation such as the European GDPR (General Data Protection Regulation) sets guidelines for personal data protection. These include sensitive data such as medical records. Medical record handling and usage require reliable procedures adhering to the legislation, as various stakeholders should have access to the data, while their ownership belongs to the patient.

Systems that store and provide access to medical data must comply with the rules in place and incorporate the concepts defined in the legislation. Ideally, such a system should implement the legislation using a high-level specification of the policies. This high-level specification should be able to represent a range of complexity ranging from simple rules, to cases where rules are overridden under specific circumstances establishing a hierarchy prioritizing rules over others, depending on the environment. This makes frameworks that can represent non-monotonic states ideal for modeling such systems. For example, in [11] the authors implement a system that models the relevant legislation of an EU country relating to medical data access. Their implementation is based on the argumentation framework Gorgias that allows for a high-level declarative representation of the policies, and is able to capture contextual-based exceptional decisions via

priority rules. However, this declarative approach is limited to their decision policy module, and does not include the interaction between stakeholders.

To this end we present a use case for medical data access that has been modeled using the rule language Prova which also supports reaction rule based workflows, event processing, and reactive agent programming. The use case includes typical stakeholders such as a patient, doctors, data controller and others. The access control is realized using the concepts of consent of the patient, the purpose of access, and the role of the party requesting access. As in [11], we define exceptions to the access rules e.g., in emergency situations. Prova has been used in the medical domain [2], as well as for modeling GDPR-compliant data wallet applications [9].

The remainder of this paper is organized as follows: Section 2 provides an overview of the Prova language. The use case is presented in Section 3, while its Prova implementation in Section 4. Section 5 discusses the results and presents related work. Finally, Section 6 concludes the paper and proposes future work.

2 Prova Basics

Prova is both a (Semantic) Web rule language and a distributed (Semantic) Web rule engine. It supports reaction rule based workflows, event processing, and reactive agent programming. It integrates Java scripting with derivation and reaction rules, and message exchange with various communication frameworks [4,2,6]. The message exchange mechanism of Prova is discussed in Section 4.

Syntactically, Prova builds upon the ISO Prolog syntax and extends it, notably with the integration of Java objects, typed variables, F-Logic-style slots, and SPARQL and SQL queries. Prolog-like compound terms can be represented as generic Prova lists (e.g., a standard Prolog-like compound term $\mathtt{f}(\mathtt{t}_1, ..., \mathtt{t}_N)$ is a syntactic equivalent of the Prova list $[\mathtt{f}, \mathtt{t}_1, ..., \mathtt{t}_N]$). Slotted terms in Prova are implemented using the arrow expression syntax '`->`'as in RIF and RuleML, and can be used as sole arguments of predicates. They correspond to a Java HashMap, with the keys limited to Stings [3].

Semantically, Prova provides the expressiveness of serial Horn logic with a linear resolution for extended logic programs (SLE resolution) [7], extending the linear SLDNF resolution with goal memoization and loop prevention. Negation as failure support in the rule body can be added to a Knowledge Base (KB) by implementing it using the cut-fail test as follows:

```
not(A) :- derive(A), !, fail().
not(_).
```

Notice the Prova syntax for `fail` that requires parentheses, as well as the built-in meta-predicate `derive` that allows to define (sub) goals dynamically with the predicate symbol unknown until run-time [6].

Prova implements an inference extension called literal *guards*, specified using brackets. By using guards, we can ensure that during unification, even if the target rule matches the source literal, further evaluation is delayed unless a guard condition evaluates to true. Guards can include arbitrary lists of Prova literals

including Java calls, arithmetic expressions, relations, and even the cut operator. Prova guards play even a more important role in message and event processing as they allow the received messages to be examined before they are irrevocably accepted. The guards are tested right after pattern matching but before a message is fully accepted, so that the net effect of the guard is to serve as an extension of pattern matching for literals [3,8]. Moreover, guard constraints can be also used to define additional constraints on metadata scopes, i.e. they can be used to define constructive views on the knowledge base by constraining the reasoning to certain selected knowledge axioms, which are selected by scoped literals on the basis of their metadata annotations (e.g., temporal, spatial, legal metadata).

3 Use Case

In this section, we present a concrete use case from the medical domain, which involves actors having to comply with established legal rules for patient data access and a situation in which the latter are overridden in the presence of an emergency situation.

The use case is built around a distributed data wallet scenario, in which a patient owns and controls a secured storage space containing sensitive personal medical data. The data wallet infrastructure allows the patient to share the data with other parties, such as doctors and hospitals.

Sharing data is accomplished by granting appropriate permissions (such as "read" and "write") and binding each permission to a particular purpose (for example, "diagnosis/treatment", "research", "emergency handling", etc.). In this scenario, the entity storing the data and providing controlled access to it by the owner as well as selected third parties, is referred to as the *data controller*, which is a term from the European General Data Protection Regulation (GDPR) [1], as are several concepts employed in this use case, such as *consent* and *purpose of consent*. For a more detailed example of a personal data wallet scenario in the context of GDPR and an in-depth introduction to the concepts and terminology involved, we refer the reader to [9].

Usually, distributed data wallet infrastructures require a complex interworking of a variety of entities, such as identity providers and multiple data controllers for each party involved. For the sake of understandability, we abstract from this degree of complexity and assume that a single data controller provides all the necessary infrastructure for data storage and access, permission handling, and authentication.

The use case description captures the dynamic behavior of the system. It consists of two interlinked parts: a description of a sequence of actions/message exchanges between actors and a description of states the system can be in, and state transitions as a consequence of an action/message. We illustrate the use case by a series of sequence diagrams, as well as a state diagram. Linking between the two is accomplished as follows: Actions that trigger a state change are marked in bold in the sequence diagrams, and state transitions carry the corresponding action names in the state diagram.

In the first part of our use case, a patient consults a doctor. They both log in to the data controller, and the doctor admits the patient to a hospital, where a medical health record of the patient is created. Figure 1 shows a sequence diagram of the above actions.

- A patient *Patient* authenticates with the data controller.
- A doctor *Doctor1* logs in to the data controller.
- Patient is admitted to a clinic by Doctor1.
- Patient demonstrates consent for health data sharing with clinical doctors at the hospital for diagnostic/treatment purposes.
- Doctor1 uploads clinical data of patient.
- Doctor1 uploads more clinical data of patient.
- Doctor1 requests all patient history.

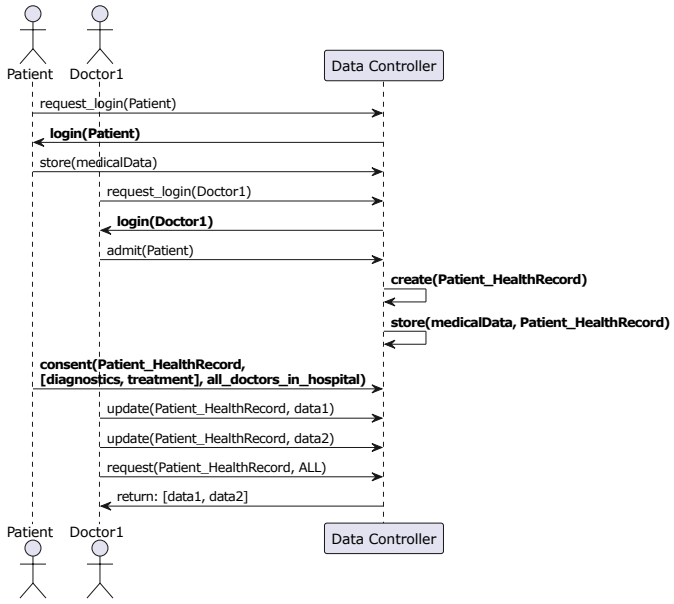


Fig. 1. The first part of the interaction of our use case

In the second part, a researcher is interested in the patient's data for a medical study. The patient consents to the use of her anonymized data for research purposes. The doctor or the researcher can prompt the hospital to produce an anonymized version of the patient's health record for research purposes, as depicted in Figure 2.

- Patient demonstrates consent for anonymized health data sharing with researchers for research purposes.

– A researcher logs in.
– The researcher requests all patient files.
– The researcher realizes that no anonymized data exist so proceeds with requesting anonymized data generation.
– Researcher requests again all patient files in her department.
– Doctor1 requests anonymized data.

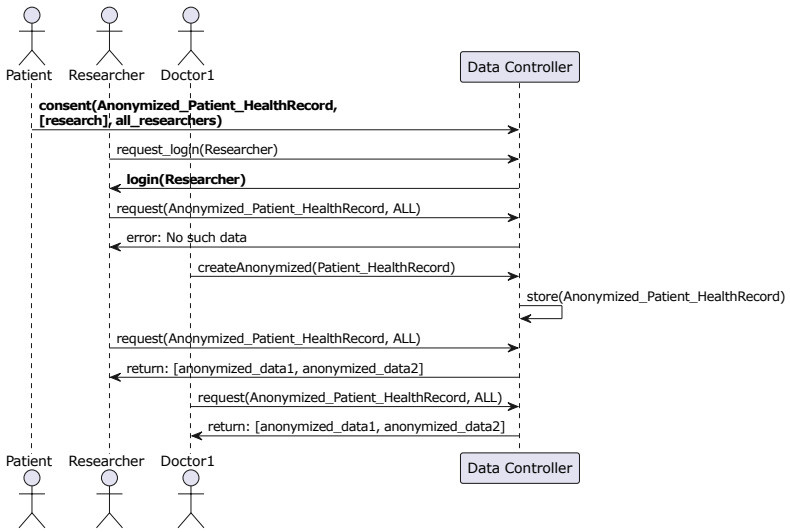


Fig. 2. The second part of the interaction of our use case

The third part describes a possible sequence of actions in an emergency situation, which temporarily overrides the existing rules for data access. During the emergency of the patient, all doctors of the hospital who are logged in to the data controller can view the patient's data for the purpose of handling the emergency. As soon as the emergency is lifted, all access permissions are reset to their previous state, as seen in Figure 3.

– Patient is cured and discharged from the clinic by Doctor1
– Doctor1 requests again all patient history
– Patient is feeling unwell and calls the emergency services
– Nearby researcher sees him in distress and requests all patient data for providing emergency assistance
– Emergency doctor *Doctor2* also comes in to help and logs in to the data controller
– Doctor2 requests all patient history for treating the patient
– Doctor2 uploads new clinical data of the patient
– Patient is now treated and Doctor2 lifts the emergency for the patient

Figure 4 shows the possible states and state transitions of our use case.

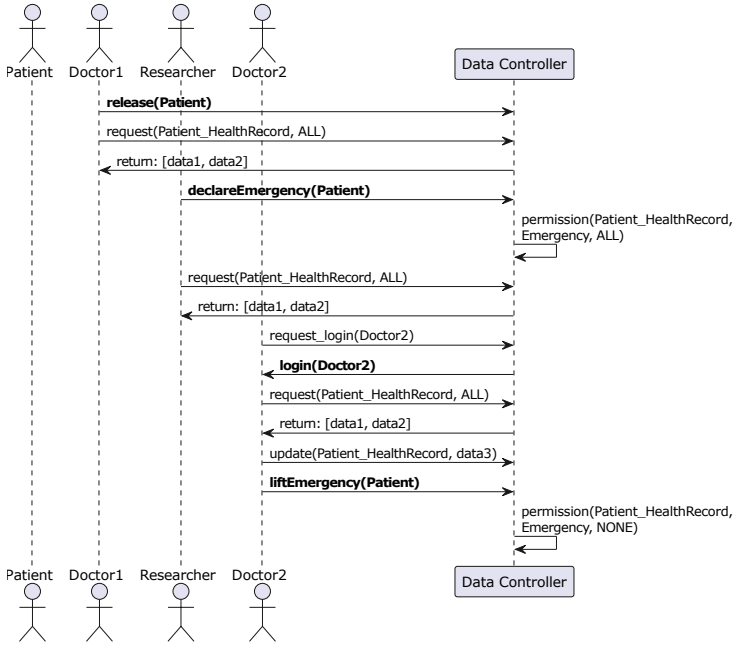


Fig. 3. The third part of the interaction of our use case

4 Implementation

The implementation[5] utilizes Prova's features such as message exchange, reactive agent messaging with assertions and retractions, and guards.

Prova's reactive agents are instances of running rulebases that include message passing primitives. In the presented use case, the following agents are defined: the `patient`, `doctor1` and `doctor2`, the `researcher`, the `dataController` and the identity provider `idp`. Both `doctor1` and `doctor2` are instances of the rulebase `doctor`, while all other agents have a dedicated rulebase.

All agent rulebases are utilizing the built-in message passing primitives to communicate, which are the predicates `sendMsg/5`, `rcvMsg/5`, as well as their variants `sendMsgSync/5`, `rcvMult/5`. The position-based arguments for the above predicates are [3]:

1. *XID* - conversation id of the message
2. *Protocol* - name of the message passing protocol
3. *Destination* or *Sender* - the agent name of the receiver/sender
4. *Performative* - the message type characterizing the meaning of the message
5. *Payload* - a Prova list containing the actual content of the message

[5] The source code is available at https://github.com/tmitsi/recomp-usecases

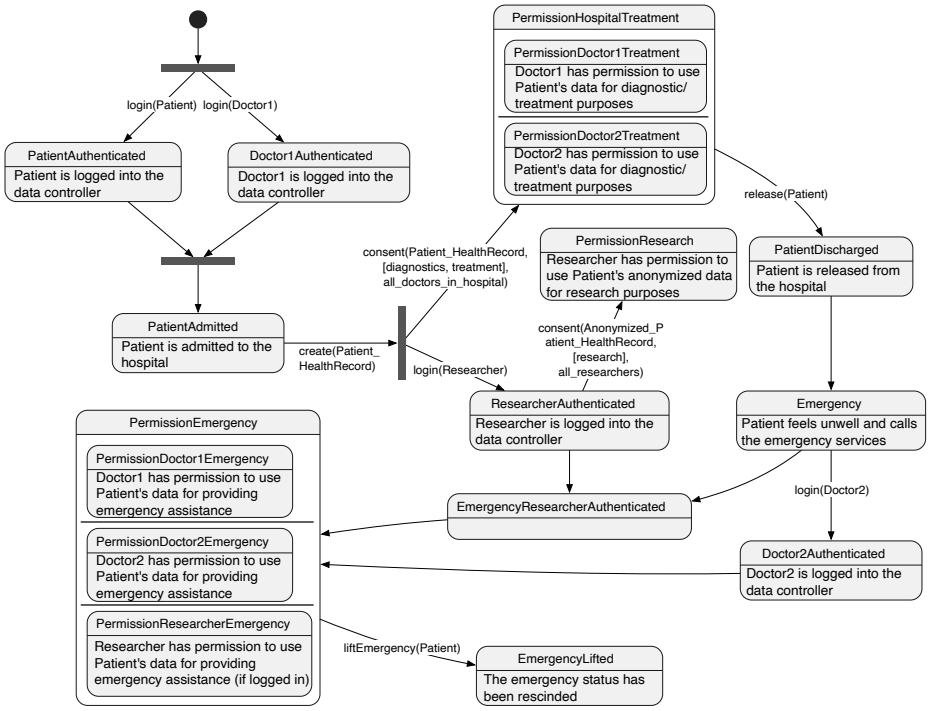


Fig. 4. The states and state transitions of our use case

Prova defines the Java interface `ProvaService` and its default implementation `ProvaServiceImpl` that allows for a runner Java class – depending on the modularization (mapping each agent to a separate bundle vs. multiple agents in a bundle) – to embed one or more agents communicating with each other via messaging. The fundamental method is the method `send`, as follows:

```
send(String xid, String destination, String sender,
     String performative, Object payload, EPService callback)
```

The arguments have a direct correspondence with the message passing primitives, while `EPService` is a superclass of the `ProvaService` interface. Also, the message passing protocol is selected automatically (in our use case, the osgi protocol is selected).

Agent actions are initiated from a Java runner class that sends a message to the agent. For example, to instruct the agent `doctor1` to log in, we first prepare the HashMap `payload`. Each key-value HashMap element corresponds to a slot (specifically, to the slot name and filler):

```
Map<String, String> payload = new HashMap<>();
payload.put("operation", "login");
payload.put("idp", "idp");
payload.put("dataController", "dataController");
```

Then we send the message, invoking the method `send`, described above:

```
service.send("xid", "doctor1", "javaRunner",
    ↪ "request", payload, this);
```

The above message is followed by an invocation of the `wait(long)` method in the service's thread, in order to give time to the Prova threads to complete the requested operations.

The Prova agent `doctor1` captures the message and forwards it to the `dataController`, expecting to receive back a message from `idp` containing a token used for subsequent requests to the `dataController`. This is accomplished by the following inline reaction rule [3]:

```
doctor() :-
    rcvMult(XID,P,javaRunner,request,
        ↪ {operation->login,
        ↪ idp->IDP,dataController->DC}),
    sendMsgSync(XID,P,DC,request,
        ↪ {operation->login,idp->IDP}),
    rcvMult(XID,P,IDP,data_exchange,{operation->assert,
        ↪ token->TOKEN}),
    assert(useToken(DC,TOKEN)),
    println(["log: doctor asserted token ", TOKEN, "
        ↪ for authenticating with ", DC]),
    spawn(XID,$Service,resume,[]).
```

Note that in the above rule the assertion happens only if `doctor1` receives back from the `idp` a message containing the generated token, as `dataController` checks if the agent is already logged in (see also Table 1). If `doctor1` receives such a message, the received token is asserted to the KB. This constitutes a positive update transition, i.e., a fact assertion [6] (retractions are similarly also called negative update transitions).

The last evaluated predicate `spawn/4` is a Prova built-in that invokes a Java method. In our case, it results in a `notifyAll()` call, signaling that the operation is complete and all threads can resume. The argument `$Service` is a global constant [3] defined in the Java runner class and in this case corresponds to the instance of the runner class.

For demonstrating consent for data sharing, the `patient` receives a message from the `javaRunner`, and makes a request to the `dataController`, specifying the operation, purpose, and parties that will be allowed to access the data. The `dataController` captures the message with the following inline reaction rule:

```
dataController() :-
    rcvMult(XID,P,A,request,
        ↪ {operation->demonstrateConsent,
        ↪ purpose->PU,parties->PA,token->T})
        ↪ [token_check(A,T)],
    assert(consent(A,PU,PA)),
```

```
    println(["log: DC asserted consent for sharing
       ↪ data with ", PA, "s for ", PU, "
       ↪ purpose(s)", " for ", A]),
    spawn(XID,$Service,resume,[]).
```

The message is accepted only if the guard `[token_check(A,T)]` succeeds. It checks if `dataController` has an agent `A` associated with the token `T`, while also checking if the variable `T` is bounded (otherwise an agent would be able to log in by sending unbounded token).

In the presented use case, the `patient` provides consent for both medical data sharing for treatment purposes with clinical doctors, as well as for anonymized medical data sharing for research purposes with researchers and doctors.

Medical data of consenting patients can be added by clinical doctors for patients admitted in their departments. In this implementation, we are using just a representation of data (a filename). Then data can be accessed by clinical doctors of the department treating the patient:

```
dataController() :-
    rcvMult(XID,P,A,request,{operation->retrieveAllData,
       ↪ token->T, subject->S, purpose->PU})
       ↪ [token_check(A,T),
       ↪ staffDB(A,clinical_doctor,DEPT),
       ↪ patient(S,DEPT)],
    findall(Data,data(S,PU,Data,ANON),L),
    sendMsgSync(XID,P,A,data_exchange,{data->L}).
```

The built-in predicate `findall/3` accumulates all solutions of a goal specified in the second argument, and returns them as a Prova list in the third argument. The first argument provides the pattern specifying the actually desired elements to be added to a list given the current goal solution [3]. As in many Prolog implementations, it does not fail if no solutions are found, instead returning an empty list.

In emergency situations, the criteria of medical data access change. In our use case, we assume that every medical personnel can request access to the medical data of a person in an emergency situation. The corresponding rule of the `dataController` capturing requests for data access in emergency situations is as follows:

```
dataController() :-
    rcvMult(X,P,A,request,{operation->retrieveAllData,
       ↪ token->T, subject->S, purpose->PU})
       ↪ [bound(S), emergency(S), token_check(A,T),
       ↪ staffDB(A, Role, DEPT)],
    findall(Data,data(S,PU,Data,ANON),L),
    sendMsgSync(X,P,A,data_exchange,{data->L}).
```

The above two rules differ only in the guarded constrains of the receiving message. The first rule allows access only for clinical doctors in the department in which the

patient is admitted: `staffDB(A,clinical_doctor,DEPT), patient(S,DEPT)`. The second rule, for emergency situations, has less restrictions on the role and department of the sender (`staffDB(A, Role, DEPT)`) constituting a more general context, but also a stricter context that defines the emergency situation (`bound(S), emergency(S)`). Thus, if patient `S` is in emergency, the default access is overridden, allowing more roles to access the data.

Additional anonymized medical data of patients that provided consent for anonymized data sharing for research purposes can be generated. They can then be accessed for research purposes by researchers and doctors.

Similarly, inline reaction rules exist for operations such as data access for research purposes, asserting an emergency, and accessing data under emergency.

Table 1 provides a high-level description of the operations defined for the agent `dataController`. The first column describes the operation (which is specified in the payload), the resulting KB update ("`+`" for a positive update transition, "`-`" for negative), and the rule functionality, while the second column specifies the sender. The third column contains the additional message payload slots. The slot `subject` is describing the patient, and the slot `parties` is defining the role. The last column provides a high-level description of the guard constrains applied to the received messages.

Table 1: High-level description of inline reaction rules of `dataController`

Operation		Sender	Slot Names	Guard Constraints
+	login	any	idp	sender not logged in
	login (displays error message)	any	idp	sender logged in
+	admit patient	clinical doctor	token, subject	doctor id verification, patient not already admitted
	admit patient (displays error message)	clinical doctor	token, subject	doctor id verification, patient already admitted
-	discharge patient	clinical doctor	token, subject	doctor id verification, patient already admitted
+	demonstrate consent	patient	purpose, parties (roles)	patient id verification
+	assert patient data	clinical doctor	data, token, subject	doctor id verification, patient consent
	retrieve patient data (rule accumulates all data with consent)	any	token, subject, purpose	sender id and role verification, patient admitted

continue on the next page

Table 1: High-level description of inline reaction rules of `dataController` (Continued)

Operation		Sender	Slot Names	Guard Constraints
+	generate patient anonymized data (rule anonymizes all data with consent for research)	any	token, purpose	sender id and role verification
	retrieve anonymized data (rule accumulates all anonymized data with consent)	any	token, purpose	sender id and role verification, patient consent
+	declare patient emergency	any	token, subject	sender id verification
	retrieve patient data in an emergency situation (rule accumulates all data with consent)	any	token, subject, purpose	patient in emergency, sender id verification, sender has medical training
+	assert patient data	emergency doctor	data, token, subject	emergency doctor id verification, patient in emergency, patient consent
-	lift patient emergency status	emergency doctor	token, subject	sender id and role verification
	retrieve patient data (displays error message)	any	token, subject, purpose	sender id and role verification, patient not provided consent
	any (displays error message)	any	token	failed id verification

5 Discussion and Related Work

The agent-based Prova implementation with the message passing primitives was able to model all key parts of the workflow. The reactive rules, combined with positive and negative update transitions were adequate to model the possible states of the use case. As shown in Section 4, the state transitions are non-monotonic, and Prova is able to express exceptions to general rules. For example, in an emergency situation, agents that under normal circumstances would not have access to a patient's medical data are provided with access.

Medical data access modeling includes the work presented in [11] using the Argumentation framework Gorgias and the tool Gorgias-B, based on the SoDA methodology. The authors model various data access levels to a patient's medical record depending on concepts such as medical service providers, patients,

controllers, and consent. The access type depends on three main contexts, namely who is asking to get access, the purpose of the access, and possible specific circumstances. Although this decision policy module is able to model the domain under consideration, the stakeholder interaction relies on external components.

Comparing our approach with the argumentation-based approach presented in [11], Prova's reactive messaging with positive or negative update transitions leads in to a sequence of states of the KB, under which the guard constraints succeed or fail. The argumentation approach, defines a contextual hierarchy of the various application scenarios from the most general to the most specific. As the presented use case does not exhibit a deep multi-level context hierarchy, Prova's guards are providing the necessary expressiveness with a relatively limited use of negation as failure mainly for printing error messages, while the use of the cut operator is avoided. Thus, both approaches are avoiding the use of red cuts [12]. Negation as failure is not used in the argumentation-based approach. Comparing the readability of the source code of the two approaches, Prova provides a better readability and is more intuitive than the pure Gorgias source code. However, Gorgias code can be generated from tools that offer similar readability levels, such as Gorgias-B or the scenario-based formalism presented in [5].

6 Conclusions and Future Work

We described a use case for medical data access and we presented its implementation in the rule language Prova. The use case focuses on medical data access, where actors such as a patient, doctors, researchers, and data controller act according to established legal rules. We also included an emergency situation, where actors can gain access, overriding their default access rights.

The implementation, realized in the rule language Prova, utilizes the concepts of consent and purpose, which are defined in the GDPR and similar legislations. These key concepts are imprinted in the Prova rules, providing a transcription of legal norms into an executable rule-based specification.

Specifically, the Prova implementation models the interaction of different parties (such as patients, doctors, and data controller), represented as agents that exchange messages with requests that can possibly assert or retract facts, resulting in different states. The reactive messaging capabilities, and the non-monotonic state transition semantics can model all possible states of the use case. The use of Prova guards provides a mechanism of defining preconditions before the acceptance of messages. This creates clear pointcuts, which are human-readable and also have the benefit that the cut operator is no longer necessary in operations with assertions and retractions.

Future work will consist in extending the implementation to a broader set of use cases. Specifically, representing legal and ethical rules side-by-side and having a reasoning procedure for deciding when an ethical rule overrides a legal rule might be of particular interest.

Furthermore, a formal evaluation of the approach using evaluation criteria of rule-based legal systems from the literature will be conducted.

As a further line of future work we plan to implement the present and possible future use cases using different representation formalisms and/or execution/reasoning environments and compare the current and future implementations as part of the evaluation. An implementation of the representational part in AspectOWL[6] [10] should be feasible and might prove advantageous wrt. usability of the modeling paradigm. In particular, AspectOWL provides mechanisms for naturally representing combinations of state and deontic knowledge that changes with state transitions (rules that apply in one state but are overridden in another state, such as an emergency). State and deontic axioms may be represented using aspects, whereas the AspectSWRL built-ins[7] permit the representation of state transitions using SWRL[8] rules. An additional advantage of AspectOWL is that it comes along with OWL's classification system[9], which can be used as a typing system for different domain concepts (such as clinical vs. legal concepts). We expect that the combination of AspectOWL as a representation/typing formalism and Prova as a message based runtime and rule execution environment might yield the best results.

Acknowledgments This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) project RECOMP (DFG – GZ: PA 1820/5-1). We also thank Gerhard Kober for the insightful discussions.

References

1. European Commission: Regulation (EU) 2016/679 of the European Parliament and of the Council (2016), http://data.europa.eu/eli/reg/2016/679/oj
2. Kober, G., Robaldo, L., Paschke, A.: Modeling Medical Guidelines by Prova and SHACL Accessing FHIR/RDF. Use Case: The Medical ABCDE Approach. In: dHealth 2022, pp. 59–66. IOS Press (2022)
3. Kozlenkov, A.: Prova Rule Language version 3.0 User's Guide (2010), https://github.com/prova/prova/tree/master/doc
4. Kozlenkov, A., Penaloza, R., Nigam, V., Royer, L., Dawelbait, G., Schroeder, M.: Prova: Rule-Based Java Scripting for Distributed Web Applications: A Case Study in Bioinformatics. In: Grust, T., Höpfner, H., Illarramendi, A., Jablonski, S., Mesiti, M., Müller, S., Patranjan, P.L., Sattler, K.U., Spiliopoulou, M., Wijsen, J. (eds.) Current Trends in Database Technology – EDBT 2006. pp. 899–908. Springer, Berlin, Heidelberg (2006)
5. Mitsikas, T., Spanoudakis, N.I., Stefaneas, P.S., Kakas, A.C.: From Natural Language to Argumentation and Cognitive Systems. In: Proceedings of the 13th International Symposium on Commonsense Reasoning. CEUR.org (2017), https://ceur-ws.org/Vol-2052/

[6] http://aspectowl.xyz/syntax
[7] https://github.com/RalphBln/aspect-swrl-builtins
[8] https://www.w3.org/Submission/SWRL/
[9] https://www.w3.org/TR/owl2-overview/

6. Paschke, A.: Rules and Logic Programming for the Web, pp. 326–381. Springer, Berlin Heidelberg (2011). https://doi.org/10.1007/978-3-642-23032-5_6
7. Paschke, A., Bichler, M.: Knowledge representation concepts for automated SLA management. Decision Support Systems **46**(1), 187–205 (2008). https://doi.org/10.1016/j.dss.2008.06.008
8. Paschke, A., Boley, H.: Reaction RuleML 1.0 for Distributed Rule-Based Agents in Rule Responder. In: Proceedings of the RuleML 2014 Challenge and the RuleML 2014 Doctoral Consortium, hosted by the 8th International Web Rule Symposium (RuleML 2014). CEUR.org (2014)
9. Schäfermeier, R., Mitsikas, T., Paschke, A.: Modeling a GDPR Compliant Data Wallet Application in Prova and AspectOWL. In: Proceedings of the 16th International Rule Challenge and 6th Doctoral Consortium @ RuleML+RR 2022, part of Declarative AI 2022. CEUR.org (2022), https://ceur-ws.org/Vol-3229/
10. Schäfermeier, R., Paschke, A.: Aspect-Oriented Ontologies: Dynamic Modularization Using Ontological Metamodeling. In: Garbacz, P., Kutz, O. (eds.) Proceedings of the 8th International Conference on Formal Ontology in Information Systems (FOIS 2014). Frontiers in Artificial Intelligence and Applications, vol. 267, pp. 199 – 212. IOS Press (2014)
11. Spanoudakis, N.I., Constantinou, E., Koumi, A., Kakas, A.C.: Modeling Data Access Legislation with Gorgias. In: Advances in Artificial Intelligence: From Theory to Practice: 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2017, Arras, France, June 27-30, 2017, Proceedings, Part II 30. pp. 317–327. Springer (2017)
12. Sterling, L., Shapiro, E.Y.: The art of Prolog: advanced programming techniques. MIT press (1994)

Danish Asylum Adjudication using Deep Neural Networks and Natural Language Processing

Satya M. Muddamsetty[1 [0000−0003−0935−4609]], Mohammad N. S. Jahromi[1 [0000−0002−6332−7567]], Thomas B. Moeslund[1 [0000−0001−7584−5209]], and Thomas Gammeltoft-Hansen[2 [0000−0003−1518−137X]]

[1] Visual Analysis and Perception Laboratory (VAP), Aalborg University, Rendsburggade 14, 9000 Aalborg
[2] Faculty of Law, Center of Excellence for Global Mobility Law, University of Copenhagen, Karen Blixens Plads 16 2300 København S, Denmark
{smmu,mosa,tbm}@create.aau.dk, tgh@jur.ku.dk

Abstract. The Danish asylum adjudication procedure is a two-tiered system, with the Immigration Service making initial determinations and the Danish Refugee Appeals Board (RAB) automatically appealing cases that are rejected. This study aims to employ a deep neural network(DNN)-based Natural Language Processing (NLP) pipeline to predict asylum decision-making outcomes using a dataset of over 15,515 Danish asylum decisions provided by the Danish Refugee Appeals Board (RAB) between January 1995 and January 2021. This research seeks to improve the performance and effectiveness of decision-making in asylum cases by addressing key challenges, such as modeling the asylum decision-making problem using NLP-based DNNs and dealing with class imbalance issues. Our preliminary results indicate that DNN-based NLP predictive models are capable of learning meaningful representations of asylum cases with high precision and recall, particularly when class weights are considered than the baseline DNN model.

Keywords: Danish Asylum adjudication · Deep Neural Networks (DNN) · Natural Language Processing (NLP) · CNN · Predictive model

1 Introduction

Europe has been at the forefront of efforts to harmonize national asylum law [10]. In particular, the Common European Asylum System and EU directives were established to standardize and streamline the procedural assessment and legal criteria for asylum claims within the European Union. Despite these efforts, however, legal outcomes for similarly situated asylum seekers continue to differ widely across individual countries. A key problem for this type of legal decision-making is that outcomes often depend on how authorities assess the credibility of asylum claims, and this assessment is prone to subjectivity and bias [3][27]. Previous studies have thus shown that the applicant's level of education, gender, and religion, as well as the decision-maker's gender, experience,

and background, can all play a role impacting decision-making [16][30][31]. In addition, several European countries have begun experimenting with AI-driven solutions to asylum decision-making [25][26][28]. While artificial intelligence (AI) models cannot yet replace the qualitative components of legal judgments, they can perform tasks such as categorization [4], identity verification [24], and data entry, and in some situations even take the role of human decision-makers in the asylum application process. For authorities, the goal may both be to reduce discrepancies, but also bolster bureaucratic efficiency in an area of ever-shifting case loads. Yet, the lack of good training data have equally generated concerns that AI models may replicate and exacerbate pre-existing human bias [6][17].
As decision-making models that rely on artificial intelligence (AI) and machine learning (ML) are becoming increasingly important in the legal domain [9][17], it's important to analyze them thoroughly. Because of their superior performance in dealing with complex data patterns and extracting meaningful insights [29], advanced ML methods like deep neural networks (DNN) are becoming increasingly plausible as large-scale datasets become available within the asylum domain. It can be extremely helpful to gain a thorough comprehension of DNN models for such cases through various evaluation metrics in order to aid in the identification of specific data patterns, improve the post-interpretation of results, and ultimately determine the potential of these tools as viable solutions for legal decision-making. If these AI-based systems are effective, they can be important resources for the legal profession, aiding in the decision-making process and improving the quality of legal decisions. This in-depth analysis allows domain experts to make informed decisions on whether to adopt and further develop AI and ML technologies, such as DNNs, in the legal domain or explore alternative approaches.

DNN has demonstrated unprecedented superior performance in a number of disciplines, including computer vision [15] and natural language processing (NLP) [20] for a number of difficult problems in these fields. However, applying DNN-NLP to the asylum domain is still in the preliminary stage. The primary objective of this study is to examine the applicability of deep learning models in the legal domain, particularly in the context of asylum decision-making. This study aims to provide valuable insight into the performance of DNN by concentrating on NLP pipelines built with DNN. Specifically, this study seeks to address the following important research questions:

1. How can DNN-NLP-based can be used to model the asylum decision-making problem?
2. What are effective methods for addressing class imbalance issues, and what is the workable strategy for developing a stable NLP-based DNN predictive model?
3. How can we judge the effectiveness of the predictive model when there is a class imbalance?

We hope to contribute to a better understanding of the application and effectiveness of deep learning techniques in the legal domain, particularly in asylum decision-making, by addressing these research questions and investigating both

general methods and a specific optimal approach. The rest of the paper is structured as follows. We review some of the recent literature where AI has been employed in the legal sphere in section 2. In Section 3 we discuss our Danish asylum dataset. Section 4 describes the proposed DNN-NLP-based asylum decision predictive modeling. Section 5 presents the experimental evaluation. Finally, Section 6 provides concluding remarks and future work.

2 Related Work

Traditional machine-learning algorithms have played an important role in the early phases of research in the field of judicial judgment prediction. In several areas of law, statistical methods such as support vector machines, decision trees, and naive Bayes classifiers have been utilized to assess and predict outcomes [6]. The study presented in [2] used support vector machines to predict rulings by the European Court of Human Rights, whereas the study in [18] relied on more conventional machine learning techniques such as decision trees and random forests to predict decisions by the United States Supreme Court. Logistic regression, naïve Bayes, and support vector machines were only some of the machine learning techniques that researchers relied on in predicting decisions in the legal domain[2] [18] [32].

In recent years, deep learning methods, especially when combined with natural language processing (NLP) methods, have gained popularity in the area of judicial decision prediction. Deep neural networks (DNN) such as convolutional neural networks (CNN), and recurrent neural networks (RNN) are just a few of the state-of-the-art techniques that have shown effectiveness in tackling complex data patterns and obtaining useful insights from legal documents. The authors in [7] employed NLP-based techniques to predict the outcome of legal decisions in English using a combination of CNNs and LSTMs. The study in [23] compares classical machine learning approaches to deep learning techniques like CNNs combined with NLP for predicting European Court of Human Rights rulings. In criminal case prediction, the authors in [22] employed deep learning techniques, such as hierarchical attention networks, in combination with NLP for predicting charges in criminal cases. This growing interest in applying deep learning methods with NLP to legal judgment prediction demonstrates their potential in offering improved performance and better capturing the complexities of legal language and case-specific information. However, the use of deep learning technologies in asylum situations is still in its early stages. As a result, the current work investigates the application of deep learning approaches for forecasting asylum decision-making outcomes in order to better understand their potential in this domain.

3 Danish Asylum dataset

The asylum decision-making procedure varies from country to country. For instance, the asylum process in Denmark is two-tiered. First-instance decisions

are made by the Immigration Service. Decisions rejected at the first instance are automatically appealed to the Refugee Appeals Board (RAB), which is a quasi-judicial body with full legal competence to assess questions of both fact and law. Denmark moreover maintains a legal opt-out to EU law, which means that it is only partly bound by common EU asylum rules [11]. In this work, we employed Danish asylum decisions provided by the Refugee Appeals Board (RAB), Denmark. The dataset used in this work consists of approximately 15,515 Danish asylum decision summaries, provided by the Danish Refugee Appeals Board (RAB), spanning from January 1995 to January 2021. The case file provides information about the applicant. The information includes country of origin, gender, religion, year of applicant entry to Denmark, ethnicity, detected divergence, previous asylum, and torture cases. It also provides information where relevant to the claim about the involvement in political parties, marital status, and military service. The case file provides a brief narrative story about the applicant's reason for seeking asylum status. Finally, the case files are closed with the legal reasoning and outcome of the RAB. This information provides the reasoning that supports the decision in the case. Furthermore, the case files also contain the candidate's interview and motivation for seeking asylum, the administrative events and the asylum process that took place since the candidate entered Denmark, and the reasoned decision by the RAB. Most documents, particularly the more recent ones, include additional documents such as the initial asylum application form and/or the interview transcript from the first instance decision-maker, the Immigration Service. As it was obtained through an agreement with the Danish Refugee Council (a Danish NGO), which regularly receives case files from the RAB, our dataset does not contain the totality of decisions by the RAB. However, our dataset is statistically representative, as the yearly recognition rates published by the RAB are consistent with the yearly recognition rates calculated on our dataset. The collected asylum decisions are presented as Word or PDF documents, and some of them are printed documents that are scanned to create PDF files. To transform these texts into a machine-readable format, optical character recognition (OCR) is used. In the aforementioned dataset, the case file consists of the decision of the applicant and legal reasoning provided by decision-makers for the reason to be granted, rejected, or sent back for further evaluation. We automatically removed the legal reasoning text with the headings "Flygtningenævnet udtaler" from the case file using a regular expression. The documents that do have the legal reasoning headings are removed. We finally have 14,987 asylum case files. Table 1 summarizes the number of case files for each individual class.

Data samples	**Granted**	**Rejected**	**SentBack**
No of Samples	2,373	12,543	71
Percentage	15%	83%	0.004%

Table 1: Total number of cases for each classes.

The dataset offers an abundance of details on Danish asylum adjudication, comprising specifics of Danish asylum legislation and practice, information on each applicant's administrative procedure, and additional data including interviews and outside evidence that aren't often available to researchers. These salient features made this dataset highly suitable for modeling asylum adjudication using NLP-based deep learning methods.

4 Methodology

The topic of asylum adjudication prediction was recently characterized by Chen et al. [8], as a binary classification challenge. In their study, 441 different judges presided over 492,903 asylum proceedings held in 336 different hearing venues throughout a 32-year span from 1981 to 2013. Similar to this, Katsikouli et al. [17] assessed the predictability of the case results based on a variety of application data, including the applicant's nationality, gender, and religion. However, all of these algorithms are modeled using manually derived features, making them unable to exploit the contextual information of case files. To address these issues, we introduce an NLP-Deep Neural Network-based asylum adjudication. Data pre-processing and algorithm development are the two primary stages of the pipeline.

4.1 Pre-processing the text

We recognize that our data contains text-based asylum case files. NLP-based AI systems aims to process and predict the outcome of an asylum judgment from textual content. To train the models, large-scale datasets are being applied to asylum-decision making. Text preprocessing involves preparing and cleaning the text into a form that is predictable and analyzable for a specific task. Preprocessing the text makes the data usable and draws attention to textual elements that an algorithm can make use of. [13]. Being initial step in any pipeline of Natural Language Processing (NLP), preprocessing and refining text can have significant impact on overall accuracy of the trained model. There are numerous steps that need to be taken, including removing punctuation, deleting whitespaces, lowering case, removing stop words, lemmatization, stemming, and tokenization.

Removing punctuation and whitespace, such as commas and full stops from the comments, eliminates redundant information from text file and maintain the size of training set lower. Stop word removal (common words are removed from the text so unique words that offer the most information about the text remain). Lemmatization and stemming are then performed to normalize the text and prepare it for further processing. Normalization is the process of converting the token into its basic form (morpheme). Inflection is removed from the token to get the base form of the word. This process helps reduce the number of unique tokens and redundancy in the data. It reduces the data's dimensionality and removes the variation of a word from the text. Finally, tokenization is applied to break the text into individual tokens to feed the classifier.

4.2 DNN-NLP-based predictive model

The problem of predicting the decisions of the Danish asylum adjudications is defined as a classification task. Our goal is to predict if the information provided in a particular case is credible enough to grant the refugee status to the applicant, based on the information provided during the interview, or if there is a violation in relation to specific Articles to reject the refugee status. We model asylum adjudication using deep neural networks (DNNs) [12]. DNNs, which are based on artificial neurons and coupled in many layers to form a network, are used to predict whether asylum status should be granted to refugees.

In recent years, Convolution Neural Networks (CNN) have demonstrated ground-breaking performance in a number of NLP tasks, including text categorization [14, 19]. The embedding layer comes first in the CNN model for natural language processing, and followed by convolutional layers. In the vector space learnt by the embedding layer, words that are similar or those appearing in related situations are closer together than words that appear in unrelated settings. The embedding layer enables us to convert each word into a fixed-length vector of a defined size. The resulting vector is dense, containing real values instead of just 0's and 1's. Word vectors' fixed length and decreased dimensions enable us to express words more effectively [21]. The embedding layer has three parameters: vocabulary size, vector length for each word, and maximum sequence length.

We propose a simple CNN-NLP model for Danish asylum adjudication. Our CNN model consists of a total of five layers. Among these, we have one embedding layer, one convolution layer, and one dense layer. A ReLU non-linearity activation function is used for every convolution layer. A global average pooling layer (GAP) is added after the high-level feature extraction convolution layer, followed by a fully connected (FC) dense layer. The input layer size for this network is equal to the maximum sequence length from the training data. The number of filters used in our network is 128 and 64, respectively. The convolution kernel size used in the model is 4. A global average pooling is applied to the last convolution layer and the training procedures are described in the following section 4.3.

4.3 Training procedure of a CNN-NLP-based asylum model

The proposed methodology is trained on our RAB dataset described in Section 3. In order to train the model, the dataset is split into 80% training, 10% validation, and 10% testing subsets of a total of 14987 cases with three classes of asylum decisions. Data pre-processing steps such as removing punctuation, white spaces, unnecessary symbols, and lemmatizing text are applied to the training, validation, and test samples. Tokenization is performed on the text after noise removal and normalization. We didn't consider removing stop-words from our data. In general, stop-words can be removed, as they are considered noise that can reduce vocabulary size. However, in our case, we didn't consider removing stopwords, as they can provide some contextual information that can affect the

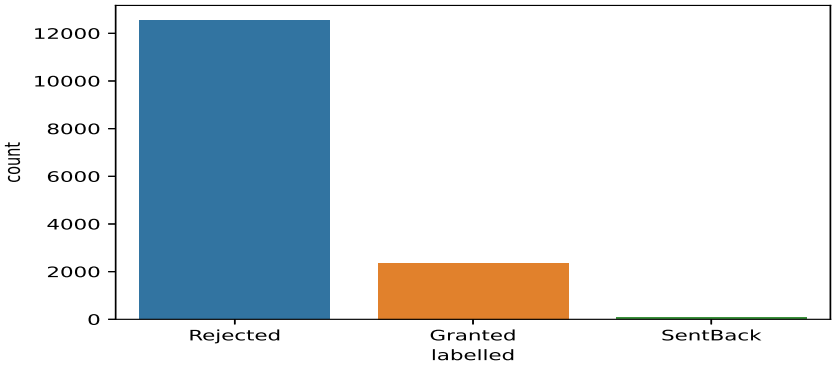


Fig. 1: Data distribution of all classes.

performance of our model in predicting the outcome of cases. The embedding layer is trained with input parameters from the full vocabulary of the trained dataset and 300 output feature vector dimensions. The CNN-NLP-based model is trained over 30 epochs with a batch size of 8 to avoid memory errors while training. We use cross entropy as a loss function and *Adam* as an optimizer, with a learning rate of 10^{-2} and momentum of 0.9. We trained our model until convergence, using an early stopping strategy that monitored the validation loss. This is a good strategy to prevent over-fitting and to save some computational time. We also added dropouts after every hidden layer to further reduce the chances of over-fitting. The framework is implemented on TensorFlow Keras with $11GB$ of GPU memory on an Nvidia, RTX 2080Ti. The predictive model is evaluated, and the results are presented in the experimental section 5.

5 Experiments & Results

In this section, we evaluate the performance of the CNN-NLP-based asylum predictive model. The proposed model is evaluated on our novel RAB dataset described in Section 3, which has three classes 'Granted',' Rejected', and 'Sentback'. We first analyzed the class distribution in our aforementioned RAB dataset. Fig 1 illustrates the distribution of cases per class and Table 1 in P.4, summarizes the number of samples for each individual class. We can clearly observe that there is a class imbalance problem in our dataset. The 'Rejected' class has higher samples followed by the 'Granted' and 'SentBack' classes with 83%, 15%, and 0.004% of total samples, respectively. In general, machine learning algorithms are not immune to unbalanced classes and generate models that are biased and less accurate. For instance, deep neural networks are trained using back-propagation, which treats each class equally when calculating the loss. If the data is not balanced, that makes the model biased for one class over another.

There are several approaches to tackle the imbalanced data problems in the literature [1]. We choose three different strategies for handling the class imbalance problem in a text dataset that contains text information. First, introducing the class weights. Class weights alter the loss function directly by penalizing the classes with varying weights. The minority class may be assigned more weight, while the majority class may be given less weight. In this manner, balance between the various classes can be achieved [5]. Second undersampling for the Majority Classes. Undersampling for the majority Class, we essentially remove a certain number of samples associated with the Majority classes for balancing the classes. Third, oversampling for minority classes, on the other hand, entails the repetition of samples associated with the minority classes. We choose three different measures recall, precision, and F1-score to evaluate the performance of our CNN-NLP-based asylum predictive model. The recall (or true positive rate) gives the proportion of the actual class correctly predicted. The precision gives the proportion of positively predicted cases that are the correct class. The F1-score gives the harmonic mean of precision and recall. We didn't consider computing the overall accuracy of the model as it can mislead when there is a class imbalance issue in the dataset.

We conducted the class imbalance experiments using the above-mentioned strategies. First, we removed the minor class, which has 71 cases which are 0.004% of the total samples, and trained the CNN-NLP model as a binary classification problem described in Section 4 on RAB dataset. Since this is the first work on the RAB dataset where a truly DNN-NLP-based model is suggested, we cannot directly compare it with the work of others. Therefore, we chose to compare the results with the proposed CNN baseline model together with different imbalance strategies. We represent the CNN model without data balancing approach as "CNN(Baseline)" in the tables. Table 2 summarizes the performance of the CNN model after applying the aforementioned class imbalanced strategies.

Binary Class Predictive Model			
Models	**Precision**	**Recall**	**F1-Score**
CNN (Baseline)	0.82183	0.84718	0.82477
CNN+ class weights	**0.82948**	**0.85188**	**0.83184**
CNN+ oversampling	0.80880	0.83847	0.81451
CNN+ downsampling	0.81126	0.75201	0.77396

Table 2: Precision and Recall for the Binary Class asylum predictions

From the tables, we can clearly observe the dealing class imbalance by introducing the class weights can significantly improve the performance of the CNN model, resulting in high precision and recall, followed by downsampling. Table 3, 4 summarizes the result on each class obtained for different imablancing strategies.

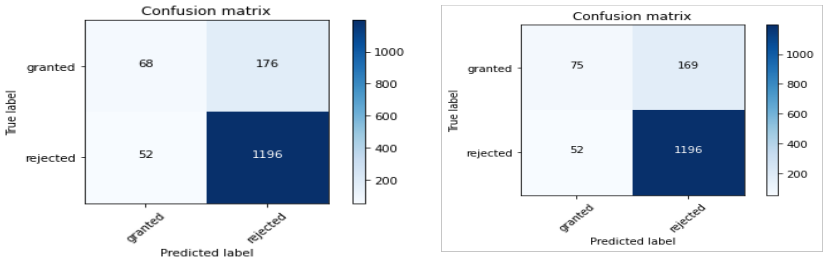
Precision for each Class	Granted	Rejected
CNN (Baseline)	0.57	0.87
CNN + class weights	**0.59**	**0.88**
CNN + oversampling	0.51	0.87
CNN + downsampling	0.34	**0.90**

Table 3: Precision table for all the multi-class experiments:

Recall for each Class	Granted	Rejected
CNN (Baseline)	0.28	0.96
CNN+ Class weights	**0.31**	**0.96**
CNN + Oversampling	0.25	0.95
CNN + Downsamping	**0.57**	0.79

Table 4: Recall for each individual class for different imbalance strategies

Analyzing the precision-recall for each individual class, we can clearly observe that the class weight strategy and downsampling strategy both demonstrate equally better precision and recall for both classes. However, downsampling strategy can balance the classes by removing the samples from the majority which might prevent the model from learning crucial information that could have been gained from the removed samples.



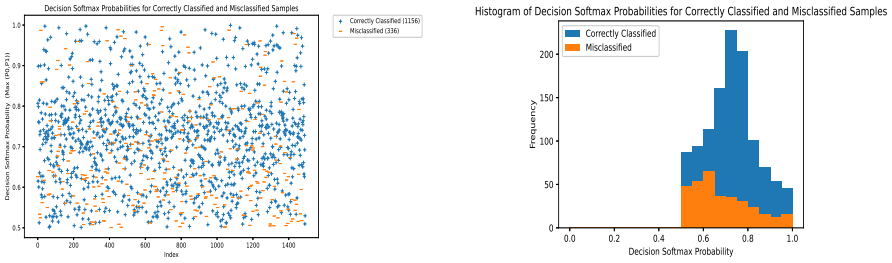
(a) (b)

Fig. 2: Confusion matrix with .(a) class weights, (b) without class weights.

The confusion matrices of both CNN models, with and without class weights, are illustrated in Fig 2. From the figure, we can clearly see the number of samples from both classes improves in correctly predicting the cases. From the tables, we conclude that addressing the class imbalance problem using class weights can improve performance in terms of precision and recall. Although either of the two strategies balances out the dataset, they do not directly tackle the issues caused by class imbalance.

We further analyzed the binary classification results of the CNN model, which performed better using the class weighting strategy. We plotted the predicted

results in terms of corrected and misclassified samples from both classes. Fig 3 shows the scatter plot and histogram distribution of correctly classified versus misclassified sample predictions of the CNN model on the test dataset.



(a) (b)

Fig. 3: Misclassified vs Classified plot. (a) Soft-max values plot, (b) Histogram distribution.

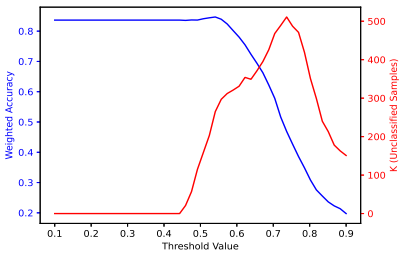
This plot illustrates the distribution of correctly classified and misclassified samples (error) across the range of decision soft-max probabilities. From this graph, we can observe that the majority of correct classes tend to have a soft-max probability greater than 0.7 (denser samples between 0.7 and 0.8). We also can observe an overlap between the correctly classified and misclassified samples of around 0.5 to 0.6 which suggests the model struggles to distinguish between classes in some cases, resulting in errors. This plot clearly explains that selecting the threshold influences the performance of the predictive model to take decisions. Hence, this suggests that traditional accuracy calculation lacks a mechanism for avoiding classifying uncertain samples, which can result in inaccurate classifications and poor performance. In addition, it implies that with a class imbalance dataset, a classifier may obtain high overall accuracy simply by predicting the majority class for all samples. In real-world situations, not all the samples need to be classified, particularly if the classifier is unsure of their class assignment. Therefore, by adjusting the threshold, when the classifier is unsure, we can choose to emphasize accuracy on classified data with higher confidence while leaving a certain number of samples unclassified, resulting in less bias in overall accuracy. This can lead to a more realistic performance calculation by further refining the accuracy of the classifier that leaves out uncertain samples and offers a great tool to control the trade-off between accuracy and uncertainty in the model.

To motivate this point, we choose to define a small epsilon region around possible range of thresholds, within which the model is undecided, as a unclassified region. Next, we compute the weighted accuracy as portion of correcty classified sample over total number of classified samples (excluding the unclassified

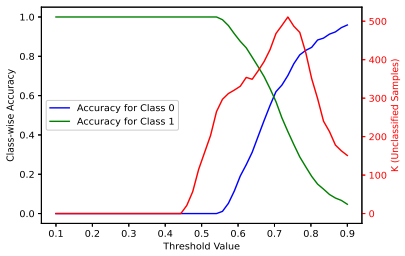
samples). Hence, the weighted accuracy of the model is defined in eq.

$$ACC = \frac{CP}{N - K} \tag{1}$$

Where CP is the total number of correct predictions for both classes "granted" and "rejected." N is the total number of input samples, and K is the number of unclassified samples (i.e., how many samples are left out after choosing the threshold value). We plotted the weighted accuracy-unclassified samples K for the two classes against the different threshold values τ ranging from 0 to 1. Weighted accuracy class-wise means computing accuracy for each class individually while taking into account the proportion of each class in the dataset. This can aid in understanding the classifier's performance in each class separately, especially when there is a class imbalance. Fig 4 (a) illustrates the weighted accuracy vs threshold values for the correct predictions. We similarly plotted each individual class in Figure 4 (b).



(a) Weighted accuracy (overall) and K against the threshold value.



(b) Class-wise accuracies for the two classes and K against the threshold value.

Fig. 4: Trade-off between weighted accuracy and class-wise accuracies for different threshold values.

From the figure, we can observe that for weighted accuracy (4. a) threshold values between 0.1 and 0.55, the weighted accuracy remains pretty stable (about 0.83). This indicates that the classifier's performance is constant within this range of threshold values. However, as the threshold value exceeds 0.55, the weighted accuracy rapidly decreases, reaching 0.2 at a threshold value of 0.9. This means that as the threshold becomes higher, the classifier's performance on classified data drops, and the number of unclassified samples mainly increases within this range. It's possible that the classifier is unsure about the class assignments for data in this range and will struggle to appropriately classify them. The class-wise accuracy case (4. b) plot shows that the classifier is very accurate for class 1 (i.e., rejected class) samples within a narrow range of threshold

values (0.1–0.55) but struggles to classify class 0 (i.e., granted class) samples. The classifier becomes more accurate in classifying class 0 (i.e., granted class) samples but less accurate in class 1 (i.e., rejected class) samples as the threshold value increases. This trend could imply a performance imbalance in the classifier between the two classes.

In practice, a threshold value should be chosen that strikes a balance between the required classification performance and the allowed number of unclassified samples while taking into account the specific needs of the target application.

6 Concluding Remarks & Future Work

In this paper, we proposed deep neural network (DNN)-NLP-based methods for modeling the Danish asylum adjudication based on textual information from asylum cases. In particular, we used the CNN model that is trained on Danish Asylum Dataset comprising approximately 15515 Danish asylum decisions provided by the Danish Refugee Appeals Board (RAB). Three distinct approaches were investigated in our research to address the class imbalance problem. We also analyzed the performance of CNN with the influence of choosing the threshold. Experimental results show that the performance of the CNN model with class weights is significantly better than the baseline CNN model when there is a class imbalance problem. Furthermore, choosing the optimal threshold for the model is crucial to lowering the misclassification rate when the dataset has a class imbalance problem. Overall, our experimental findings in this work points to the existence of further key issues that are yet to be addressed in this application context. This is mainly due to the limitation of accessing large target domain datasets to train with deep learning models that require millions of parameters. This will also limit the model's capacity to create domain-specific pre-trained word embeddings, which act as a pillar for training the model to deliver improved accuracy. However, we believe that by demonstrating the great potential of using deep learning in the legal domain, specifically in their application to decision-making or similar prediction cases, we can encourage further application and innovation of these models in the legal domain that provides a great value to the research community. To further improve performance, as future work, we intend to collect large data multilingual asylum dataset with focus on training with pre-trained word embeddings. In addition, we aim to investigate interpretibility of the predictive models using explainable AI (XAI) framework, which can help to gain deeper insight of their decision-making process.

7 Acknlowledgements

This work is part of the 'Explainable Artificial Intelligence and Fairness in Asylum Law (XAIfair)' interdisciplinary project, funded by The Villum fonden, Denmark.

References

1. Abd Elrahman, S.M., Abraham, A.: A review of class imbalance problem. Journal of Network and Innovative Computing **1**(2013), 332–340 (2013)
2. 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 Computer Science **2**, e93 (2016)
3. Anker, D.E., Müller, M.: Explaining credibility assessment in the asylum procedure. International Journal of Refugee Law **19**(3) (2007)
4. Bayer, M., Kaufhold, M.A., Reuter, C.: A survey on data augmentation for text classification. ACM Computing Surveys **55**(7), 1–39 (2022)
5. Byrd, J., Lipton, Z.: What is the effect of importance weighting in deep learning? In: International conference on machine learning. pp. 872–881. PMLR (2019)
6. Byrne, W.H., Gammeltoft-Hansen, T., Piccolo, S., Møller, N.L.H., Slaats, T., Katsikouli, P., et al.: Data driven futures of international refugee law (2021)
7. Chalkidis, I., Kampas, D.: Deep learning in law: early adaptation and legal word embeddings trained on large corpora **27**, 171–198 (2019). https://doi.org/10.1007/s10506-018-9238-9, https://doi.org/10.1007/s10506-018-9238-9
8. Chen, D.L.: Judicial analytics and the great transformation of American Law **27**, 15–42 (2019). https://doi.org/10.1007/s10506-018-9237-x, https://doi.org/10.1007/s10506-018-9237-x
9. Chen, D.L., Eagel, J.: Can machine learning help predict the outcome of asylum adjudications? In: Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law. pp. 237–240 (2017)
10. Fry, J.D.: European asylum law: Race-to-the bottom harmonization. J. Transnat'l L. & Pol'y **15**, 97 (2005)
11. Gammeltoft-Hansen, T., Scott Ford, S.: An Introduction to Danish Immigration Law. SSRN Electronic Journal (234) (2021). https://doi.org/10.2139/ssrn.3769962
12. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
13. Hasanah, U., Astuti, T., Wahyudi, R., Rifai, Z., Pambudi, R.A.: An experimental study of text preprocessing techniques for automatic short answer grading in indonesian. In: 2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE). pp. 230–234. IEEE (2018)
14. Hassan, A., Mahmood, A.: Convolutional recurrent deep learning model for sentence classification. Ieee Access **6**, 13949–13957 (2018)
15. He, K., Zhang, X., Ren, S., et al.: Sun., j.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
16. Johannesson, L.: In courts we trust: Administrative justice in Swedish migration courts. Ph.D. thesis, Department of Political Science, Stockholm University (2017)
17. Katsikouli, P., Byrne, W.H., Gammeltoft-Hansen, T., Høgenhaug, A.H., Møller, N.H., Nielsen, T.R., Olsen, H.P., Slaats, T.: Machine learning and asylum adjudications: From analysis of variations to outcome predictions. IEEE Access **10**, 130955–130967 (2022)
18. Katz, D.M., Bommarito, M.J., Blackman, J.: Predicting the behavior of the supreme court of the united states: A general approach. SSRN Electronic Journal (2014)
19. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D.: Text classification algorithms: A survey. Information **10**(4), 150 (2019)

20. Lee, J., Toutanova, K.: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
21. Levy, O., Goldberg, Y.: Dependency-based word embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 302–308 (2014)
22. Luo, B., Feng, Y., Xu, J., Zhang, X., Zhao, D.: Learning to predict charges for criminal cases with legal basis. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 2727–2736 (2017)
23. Medvedeva, M., Vols, M., Wieling, M.: Using machine learning to predict decisions of the european court of human rights. Artificial Intelligence and Law **28**(2), 237–266 (2020)
24. Mohammed, I.A.: Artificial intelligence for cybersecurity: A systematic mapping of literature. ARTIFICIAL INTELLIGENCE **7**(9) (2020)
25. Molnar, P., Gill, L.: Bots at the gate: A human rights analysis of automated decision-making in canada's immigration and refugee system (2018)
26. Nielsen, T.R.: Confronting asylum decision-making through prototyping sensemaking of data and participation. In: Proceedings of 19th European Conference on Computer-Supported Cooperative Work. European Society for Socially Embedded Technologies (EUSSET) (2021)
27. Noll, G.: Salvation by the grace of state? explaining credibility assessment in the asylum procedure. In: Proof, evidentiary assessment and credibility in asylum procedures, pp. 197–214. Brill Nijhoff (2005)
28. Ozkul, D.: Automating Immigration and Asylum: The Uses of New Technologies in Migration and Asylum Governance in Europe. Oxford University Press (2023)
29. Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M.P., Shyu, M.L., Chen, S.C., Iyengar, S.S.: A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR) **51**(5), 1–36 (2018)
30. Rottman, A.J., Fariss, C.J., Poe, S.C.: The path to asylum in the us and the determinants for who gets in and why. International Migration Review **43**(1), 3–34 (2009)
31. Spirig, J.: Like cases alike or asylum lottery? Inconsistency in judicial decision making at the Swiss Federal Administrative Court. Ph.D. thesis, University of Zurich (2018)
32. Sulea, O.M., Zampieri, M., Malmasi, S., Vela, M., Dinu, L.P., van Genabith, J.: Exploring the use of text classification in the legal domain. arXiv preprint arXiv:1710.09306 (2017)

The Argumentation Scheme from Vicarious Liability*

Davide Liga[1 [0000−0003−1124−0299]]

University of Luxembourg, Esch-sur-Alzette, Luxembourg
davide.liga@uni.lu

Abstract. In this paper, we propose the formalisation of a new argumentation scheme for the domain of legal argumentation, which we call Argument from Vicarious Liability. This scheme is particularly frequent in the domain of Tort Law and describe the concept of *Respondeat Superior*, according to which the liability of a wrongdoing can be connected to the agent who is hierarchically above the wrongdoer. While pointing out the need to deepen the study of liability in argumentation schemes and legal argumentation, this work is also proposing the first argumentation scheme which is explicitly related to liability and, indirectly, to causality, showing its connection with pre-existing argumentation schemes.

Keywords: Argumentation Schemes · Liability · Legal Knowledge Representation.

1 Introduction

In recent years, an increasing interest has been dedicated to argumentation schemes, a theoretical construct which is employed in structured argumentation, i.e. the branch of argumentation which considers arguments not as atomic structures (like for abstract argumentation), but as having an internal structure. This internal structure usually depends on which kind of *model of argument* is employed, for example the Toulmin's model of argument (composed of claim, ground, warrant, backing, qualifier, rebuttal) [11] or the Walton's model (composed of a set of premises and a conclusion [15]). Many works on Argumentation and Artificial Intelligence (AI) have focused on the Walton's model, showing its usefulness in the field of AI, especially with regard to the use of argumentation schemes [8][2][4][5]. Argumentation schemes can be described as stereotypical inferential patterns of reasoning [9], represented in natural language and showing some sort of inferential steps which people commonly employ in communication, particularly when arguing. A famous example of argumentation scheme which is stereotypically employed by people in everyday argumentation is the so-called argument from negative consequences (Table 1).

* The author was supported by the project INDIGO, which is financially supported by the NORFACE Joint Research Programme on Democratic Governance in a Turbulent Age and co-funded by AEI, AKA, DFG and FNR and the European Commission through Horizon 2020 under grant agreement No 822166

Premise	If A is brought about, bad consequences will plausibly occur
Conclusion	Therefore, A should not be brought about.

Table 1. Structure of the argument from negative consequences, which is composed of one single premise and a conclusion.

This scheme is a very common stereotypical inferential pattern, which one can easily find in everyday conversations. Broadly, these stereotypical schemes are useful to analyze and evaluate arguments in a wide range of fields, including philosophy, law, and AI. They can be seen as argumentative templates [6], which can then be instantiated in more specific ways, depending both on how natural language is used by the arguers, and on the context. In general, these schemes provide a way to identify and analyze the structure of an argument and to determine whether it is a strong or weak argument, also with the use of the so called critical questions, which can be thought as "stress tests" for the hold of the argument's structure.

While a compendium of some of the most frequent argumentation schemes has been proposed by Walton in his famous works [15], a recent work proposed a similar effort in the field of legal argumentation, therefore focusing on legal argumentation schemes [14]. The motivation behind our work is to contribute to this area of research focusing on some aspects which we consider worth deepening. In particular, we noticed that little account has been given to the role of liability in legal argumentation schemes, including the interplay between causal responsibility and legal responsibility.

In this work, we will show an example of legal argumentation scheme where legal responsibility (i.e., liability) is at stake, namely the argument from vicarious liability. As we will see, this scheme also offers a starting point for another research direction, showing the need to find argumentation schemes based on legal and causal responsibility.

In [14], we can find argument schemes which are specific of the legal statutory interpretation, for example the argument from precedent, or the argument from the application of a rule. We argue that the direction undertook by this compendium of legal argumentation schemes could be further enriched by adding an account of how causality and liability are instantiated in legal schemes. In fact, although causal responsibilities and legal responsibilities are very much important in legal reasoning (which can be channeled by the concepts of *causality* and *liability*), it seems that little account has been given to these two crucial aspects, apart from very few works (e.g., [1]). Therefore, we believe that we need to give some account of *legal argumentation schemes from causality* and *legal argumentation schemes from liability*, which we believe is currently missing. To stimulate these research directions and to show that some important argumentation schemes could fall under these two umbrellas, we describe and formalise a quite important argumentation scheme which directly take into account liability (and, indirectly, also causality). On the one side, our aim is to stimulate further research in this direction of searching for legal argumentation schemes

from causality and *from liability* (we will target the second ones). On the other side, we want to point out that this direction is not linear and the well-known difference (and overlapping) between liability and causality, deserves to have an account in legal argumentation (and in the formalisation of legal argumentation schemes). In this work, we start investigating this direction by offering a first case study which explicitly focuses on a legal argumentation scheme *from liability* (while also taking into account elements of causality). More specifically, we will refer to the idea of vicarious liability, which is often used by judges to support (or attack) argumentative standpoints in legal reasoning. To show the importance of this scheme, we will refers to two famous judgements where it was employed: *Mohamud v WM Morrison Supermarkets plc [2016] UKSC 11, 2 March 2016]* and *Cox v Ministry of Justice [2016] UKSC 10, 2 March 2016.* We will thus provide a formalisation of the scheme, including critical questions, showing how judges employed this scheme.

While describing the argument from vicarious liability, we will also suggest potential argumentative and ontological connections between our proposed scheme and existing Waltonian schemes. This is important because we believe that more account should been given to how the original Waltonian compendium and the *legal compendium* in [14] are ontologically related. This means showing, for example, how legal schemes are related to non-legal ones, or how legal schemes establish supporting/attacking relations with the original Waltonian schemes (we will briefly explore this second option).

Furthermore, it is worth mentioning that our scheme from vicarious liability is a perfect example to show that the relation between causality and liability deserves more attention from the argumentative point of view. In this regard, while showing that causal responsibility and legal responsibility does not necessarily coincide, we would also like to suggest that there might be complex (but stereotypical and thus frequent) interactions between these two spheres, which might have corresponding argumentation schemes.

To the best of our knowledge, this is the first work which deals with a specific case of legal argumentation scheme from liability, and we hope that further research can go towards this direction, shedding more light on how the interplay between liability and causality is instantiated in legal argumentation. In Section 2, we introduce the idea of vicarious liability, briefly describing its meaning and doctrine. In Section 3, we will formalize the argumentation scheme from vicarious liability before introducing two famous cases in the two sections after. In this section, we will also show the relationship with other schemes and give a brief account of how causality and liability interact in the case of vicarious liability. In Section 4, we will describe the case *Mohamud v WM Morrisons Supermarkets.* In Section 5, we will describe the case *Cox v Ministry of Justice.* In Section 6, we will discuss some aspects of the proposed scheme, while Section 7 will conclude.

2 Vicarious Liability

The Vicarious Liability is a well-known kind of liability in legal theory. This kind of liability is related to the doctrine of *Respondeat superior* (from Latin: "let the master answer"), according to which a party is responsible for acts performed by other agents. For example, in some circumstances, an employer can be liable for the actions of employees, if these actions are performed in the course of the employment. This kind of rule is sometimes referred to as "master-servant rule" and exists in both civil law and common law juridical systems.

Vicarious Liability, in a broader sense, is a form of strict and secondary liability. *Strict liability* arises when a person is considered legally responsible for the consequences of an activity even in absence of fault or criminal intent. *Secondary liability* arises when a party materially facilitates, induces, or contribute to directly infringing acts carried out by other parties. In other words, strict and secondary liabilities produces legal responsibility also in absence of a direct causal connection between the wrongdoing and the person who is target of the liability. In the more specific case of *vicarious liability*, the liability is channeled from the wrongdoer to the liable person because of the kind of relationship that exist between the wrongdoer and the liable person (e.g., an employer-employee relationship), as well as because of the context in which the wrongdoing took place (i.e., the wrongdoing must have been done in the course of such kind of relationship).

More precisely, the doctrine of vicarious liability provides that an employer can be held liable for civil wrongdoings committed by his employees if a connection *between the employer and the primary wrongdoer* (i.e., the connection, or relationship, between the employer and the employee) exists and the connection *between the employment and the wrongdoing* is sufficiently close to make it fair to hold the employer liable for the wrongdoer's actions (i.e., if the wrongdoing is sufficiently considered as something occurred in the context of the above-mentioned relationship). In other words, establishing a vicarious liability is a *two-stage test*. The first limb of the test is establishing the existence of the relationship between the wrongdoer and the vicariously liable person. The second limb of the test is establishing whether the wrongdoing occurred in the context of such relationship. This two-stage test comes from the famous case of *Lister v Helsey Hall [2001] UKHL 22* and has since been used in a wide range of legal decisions.

Clearly, this two-stage test is in itself open to legal interpretation since it inevitably comes down to a value judgement based on the particular contextual circumstances in any given case. This is where legal reasoning and legal argumentation enter the scene. In this regard, on the one side, some cases have focused more on establishing the first limb of the test, i.e., the assessment of the kind of relationship, like the case *Cox v Ministry of Justice*. On the other side, some cases have focused more on establishing the second limb of the test (sometimes referred to as "close connection" or "sufficient connection" test), like the case *Mohamud v WM Morrison Supermarkets plc*. Before analysing these

two famous cases, we will propose, in the next section, a model for the argument from vicarious liability.

3 Argumentation Scheme from Vicarious Liability

This scheme is quite frequent in tort law and is based on the above-mentioned idea of vicarious liability, according to which a the legal responsibility (i.e., the liability) of a wrongdoing is channeled from the wrongdoer (who has the direct or causal responsibility of the wrongdoing) to a second agent who has a relevant hierarchical relationship with the wrongdoer (e.g., a relationship employer-employee). As mentioned before, the assessment of vicarious liability in tort law has been traditionally assessed through a simple test consisting of two questions: (1) is there a relevant relationship between the wrongdoer and the third party (e.g., an employment relationship)? (2) is the connection between such relationship and the wrongdoing sufficiently close?

From an argumentative point of view, one can interpret this questions as the critical questions that judges must verify to assess a potential vicarious liability. In fact, to understand better, we can switch to an argumentative perspective by saying that the aim of the two-stage test is to assess the hold of the argument according to which it is the case that the doctrine of vicarious liability is applicable. Even though switching to this argumentative perspective makes it easier to see the critical questions behind the two-stage test, we still do not have a formal structure for the scheme from vicarious liability to which these critical questions are referred. In other words, what is the argument that this two-stage test tries to stress-test? First of all, an argument from vicarious liability must consider that a wrongdoer is responsible for a wrongdoing (first premise). Secondly, one need to consider not just the wrongdoer, but a second person who has a specific (i.e., relevant) relationship with the wrongdoer (second premise). Furthermore, the wrongdoing must be located in the context of such relationship (third premise). Finally, the conclusion must be that the second person is vicariously liable for the wrongdoing. Hence, we propose to design the argumentation scheme from vicarious liability as follows:

(P1) Agent A is causally responsible for wrongdoing W through Action D **[Causal Responsibility Premise]**
(P2) Agent L has a relevant vicarious relationship R with Agent A **[Vicarious Relationship Premise]**
(P3) Action D occurred in the scope of relationship R **[Vicarious Wrongdoing Premise]**
(C) Therefore, Agent L is vicariously responsible for wrongdoing W **[Conclusion]**

(CQ1) Is it really the case that there is a relevant relationship between Agent A and L? **[Vicarious Relationship Critical Question]**
(CQ2) Is the connection between relationship R and wrongdoing W sufficiently close? **[Sufficient Connection Critical Question]**

(CQ3) Is Agent A causally responsible for wrongdoing W? **[Causal Responsibility Critical Question]**

As can be seen above, the critical questions reflect the two-stage test, while adding a further critical question to stress test the very first premise. In fact, other studies have shown that each premise can be attacked on the bases of the content it provides, and premises generally provide the argument structure with their own piece of semantic information (the semantic link by which each premise give support the underlying inferential process towards the conclusion), which can have different natures (e.g., causal, factual, definitional) [7].

In general, we can consider arguments as structures which may be attacked (or supported) at specific critical points of their structure. In this regard, we can say that the hold of an argumentation scheme has a minimum amount of critical positions which corresponds to the number of premises (because each premises can be questioned) plus *at least* one critical point for the inferential step connecting the premises to the conclusion, plus at least one rebuttal stating the negation of the conclusion. In this regard, the argument from vicarious liability can be represented as having four basic critical points or critical positions (see Figure 1).

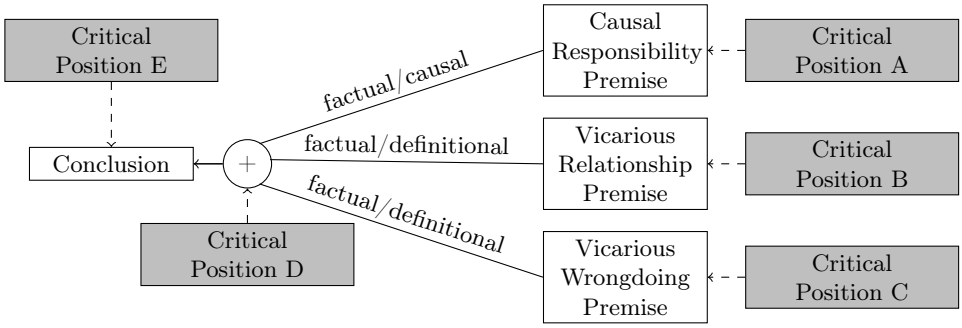


Fig. 1. Structure of the Argument from Vicarious Liability, showing the semantic link connecting the premises to the conclusion. Dashed connections are potential attacks or supports heading towards the critical positions of the scheme.

Critical Questions are positioned at critical position A, B and C. At these positions, other arguments might give an attack (or a support), which will directly attack (or support) the semantic link provided by the corresponding premise.

The first premise, for example, can be attacked or supported by a causal argumentation scheme. Instead, at position B and C the support or attack to the hold of the argumentative structure can come from a verbal classification scheme, since the premises' semantic information is mostly dedicated to factual/definitional elements (namely the nature of the relationship between wrongdoer and the potentially liable agent, as well as the nature of the scope under

which the wrongdoing occurred). At position D, attacks/supports are meant to target the inferential steps from premises to conclusion. This also includes the attack to the structure as a whole. In other words, attacks at critical position A, B and C are referred to the so-called *undermining* attack, while an arrack at position D refers to the *undercut* attack. Finally, we also mentions critical position E, the position for the *rebuttal* attacks, which occur when the conclusion is directly attacked by another argument.

Relationships with other schemes As mentioned before, the first critical position can be "stress-tested" when the first premise is attacked (or supported). Having a causal semantic link, this critical position is physiologically prone to be attacked (or supported) by causal schemes [15, 3] such as the argument from cause to effect, the argument from correlation to cause. Regarding the other two premises, they can be stressed (or supported) by the use of a verbal classification schemes [12], which are meant to further specify, semantically, the nature of the involved relationship or the scope under which the wrongdoing occurred (since these two premises can be attacked/supported mostly with respect to the definition of the two key concepts they convey, namely the words "relevant" and "scope").

Another important aspect which is worth mentioning is that we see this scheme from vicarious liability as a part of a more general family of schemes from liability, which are not yet available in literature, and whose internal relationship should be explored on the basis of the existing legal theory. In this regard, we assume that an argument from vicarious liability is likely to be a descendant of two "parent schemes" belonging to the family of the arguments from liability, namely the argument from strict liability and the argument from secondary liability. While further research is needed to explore this ontological classification, a potential classification of these schemes might look like the scheme in Figure 2.

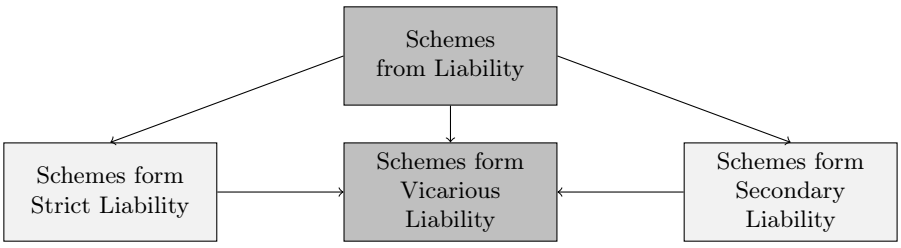


Fig. 2. Classification of some of the envisaged schemes from liability.

Interactions between liability and causality in legal argumentation

Although this paper does not aim at describing the complex interplay between liability and causation [16, 10], we argue that more efforts are required to clarify some of the main interactions between these two spheres from the argu-

mentative point of view, providing an account of how this interplay is represented in different schemes from liability.

From this point of view, the scheme from vicarious liability offers a privileged perspective, because it is located in the context of strict liability, where liability arises even if the person who is targeted by liability is not aware of the wrongdoing for which he is considered liable. This kind of liabilities may be considered, from an argumentative point of view, as belonging to what we call *argumentation schemes from strict liability* (as depicted in Figure 2). From a more abstract theoretical point of view, they dramatically remind us the necessity to keep in mind the difference between causal and legal responsibilities when evaluating wrongdoings.

In the case of our scheme from vicarious liability, it seems that liability can be generated from a fact even if there is no causal link between the person considered vicariously liable and the wrongdoing. This has the effect of extend somehow the scope of the liability for that wrongdoing, creating a "transfer" of liability from the wrongdoer (whose liability coincide with the actual causation) to the agent who is hierarchically responsible for the wrongdoer's actions if these actions occur in the context of the relationship between the two agents (and this "extended" liability thus goes beyond actual causation). It would be interesting to see what kind of potential interactions exist from the argumentative point considering different types of schemes from liability. In other words, it would be interesting to explore argumentation schemes as tools for understanding the interactions between liability and causality in legal reasoning. We leave this to future research.

4 Mohamud v. WM Morrison Supermarkets

The case *Mohamud v VM Morrison Supermarkets* is a very famous example of vicarious liability. The case was focused on the interpretation of the second limb of the above-mentioned two-stage test, which is sometimes referred to as "close connection" or "sufficient connection" test.

Facts. A man named Mr. Mohamud, who is of Somali descent, pulled over at a Morrison petrol station. The station's employee, Mr. Khan, was working at the kiosk and had the responsibility of serving customers and ensuring the proper functioning of the petrol pumps and the kiosk. Mr. Mohamud went inside the shop to inquire about printing some documents, to which Mr. Khan responded with swear words. Upon objecting to being sworn at, Mr. Khan ordered Mr. Mohamud to leave and used foul and racist language. Mr. Mohamud left the shop, got back in his car, and was about to drive away when Mr. Khan approached him, opened the passenger door, and told him never to return to the petrol station. When Mr. Mohamud asked Mr. Khan to step out of the car, he punched him in the head. Mr. Mohamud got out of the car to close the passenger door, but Mr. Khan continued to attack him, striking him and kicking him until he fell to the ground. Despite the efforts of his supervisor to stop him, Mr. Khan carried out the attack. As a result of the assault, Mr. Mohamud filed a personal

injury claim against Morrison, raising the question of whether the company was vicariously liable for Mr. Khan's violent actions.

County Court decision. The Court ruled that vicarious liability could not be established as the "close connection" test was not satisfied. The trial judge was unable to determine that the company was vicariously responsible for Mr. Khan's actions. In evaluating the second aspect of the vicarious liability test and applying the "close connection test" from the Lister case, the judge was unable to establish a sufficient connection between Mr. Khan's employment and the unprovoked assault. While it was acknowledged that Mr. Khan's job entailed some customer interaction, serving and assisting them, this was not deemed "sufficiently closely connected" to warrant holding the company vicariously liable for the attack. Another key factor in the trial judge's decision was that Mr. Khan had taken a deliberate action by leaving the kiosk and pursuing Mr. Mohamud onto the forecourt, going against his employer's instructions.

Court of Appeal decision. Mr Mohamud appealed the first instance decision but the Court of Appeal upheld the decision. The reasoning was similar to the first instance decision but went further by stating that, since Mr. Khan's responsibilities did not involve a high likelihood of conflict, merely having interaction with customers in his role was not enough to make his employer vicariously liable for his violent behavior.

Supreme Court decision. Mr. Mohamud took his case to the Supreme Court and asked for the "sufficient connection" test to be replaced with a "representative capacity" test. This proposed test was broader and asked whether a reasonable observer would consider the employee to be acting in a representative capacity for the employer at the time the tort was committed. This focus would not be on the closeness of the connection between the employee's work and the tortious conduct, but would relate to the setting the employer created. Mr. Mohamud argued that the "representative capacity" test was met as Mr. Khan, an employee responsible for serving customers at the petrol station, was the human representative of the employer and the employer created the setting by placing Mr. Khan in close physical contact with him. However, the Supreme Court rejected the "representative capacity" test, considering it unnecessary as it did not differ substantially from the Lister test. The judges preferred the broad application of the "close connection" test, which considered Mr. Khan's violent act to be sufficiently closely connected to his employment for Morrison to be vicariously liable. One of the judges argued that Mr. Khan leaving the kiosk to follow Mr. Mohamud to his car did not break the connection, stating that it would not be fair to say that Mr. Khan had "taken off his uniform metaphorically" when he stepped out from behind the counter. Therefore, the Supreme Court upheld Mr. Mohamud's claim and determined that Morrison was vicariously liable for Mr. Khan's actions.

Argumentative analysis The first judge refuted the argument of the appellant according to which Morrison Supermarkets were vicariously liable because the second question (related to the "close connection") was answered negatively. The trial judge reached this conclusion by using the critical question of the

sufficient connection (i.e., the "close connection test" from the case of Lister v Hesley Hall), to which he answered negatively: according to the trial judge the connection between the wrongdoing and the relationship is not sufficient, i.e., the wrongdoing did not happen in the sufficiently within the employer-employee relationship. The judge supported this conclusion that the sufficient connection was not met by adding that Mr Khan went out of the shop against the instructions of his employer. Once the judge undermined the third premise by answering negatively to the sufficient connection critical question, the inferential step from the premises to the conclusion was not acceptable.

The second judge presented a similar reasoning, using the critical question of the sufficient connection to affirm that the the wrongdoing did not happen within the scope of the employer-employee relationship. This time the judge used an even stronger argument by stating that Mr Khan responsibilities did not include the likelihood of a conflict and the mere confrontation with a customer does not justify the vicarious liability of the defendant. Again, this meant that the inferential passage from the premises to the conclusion of the argument from vicarious liability was rejected, because the third premise was undermined.

The last judge, instead, answered positively to the sufficient connection critical question. Therefore granting the inferential passage from the three premises to the conclusion. It is interesting to note that the appellant, coming from two negative decisions on the sufficient connection critical question, proposed to replace such critical question with a new one, in order to facilitate the inferential passage to the conclusion of the argument from vicarious liability.

Figure 3 summarises the inferential steps that judges undertook with respect to the argument from vicarious liability.

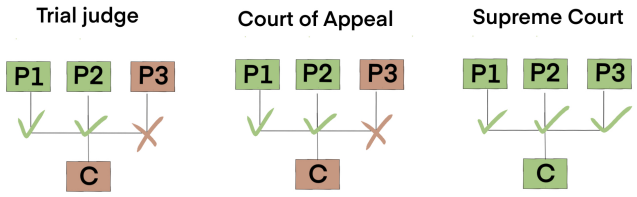


Fig. 3. Inferential processes of the three judicial level of judgment for the case Mohamud v WM Morrison supermarkets with respect to the argument from vicarious liability.

5 Cox v. Ministry of Justice

If in *Mohamud v WM Morrison Supermarkets* the Supreme Court decided about the second limb of the vicarious liability two-stage test, i.e., about the connection

between the vicarious relationship and the tortious act, in *Cox v Ministry of Justice*, the Court decided instead about the first limb of the two-stage test, which is related to the relationship between the defendant (the potential vicariously liable person) and the wrongdoer.

Facts. Mrs Cox was the catering manager at HMA Swansea and had responsibility for the kitchen operation. She supervised 4 employees and 20 prisoners. During a kitchen supplies delivery, a prisoner dropped a sack on her back while trying to carry two past her, causing injury. The incident was deemed negligent. Mrs Cox claimed that the Ministry of Justice was vicariously liable for the actions of the prisoner orderly and sought compensation for her injuries. The problem here was to assess whether a relevant relationship existed between the defendant (Ministry of Justice) and the wrongdoer. Prison rules state that convicted prisoners in state or private prisons must do useful work for up to 10 hours a day. The defendant's policy is that work instills a hard-working ethos and teaches vocational skills. Prisoners can apply to work in prison kitchens and are selected after assessments. They may be paid £11.55 per week by the Secretary of State to encourage participation. Without prisoner work, the prison service would need to incur additional costs for staff or contractors. Judges reasoned about these elements to assess whether the vicarious relationship critical question was positive. Is the relationship between the Ministry of Justice and the prisoner relevant to accept vicarious liability?

County Court decision. The trial judge ruled that the prison service was not vicariously responsible for the prisoner's negligence. He evaluated if the connection between the prison service and the prisoner was similar to that of an employer and employee and found it was not. He acknowledged that there were similarities, but noted a crucial difference. Employment is a mutual agreement where each party benefits. With prisoners, the situation is different. The prison is legally obliged to provide work and pay for it, not as a choice but as part of their penal policy. The work is meant for the prisoner's discipline, rehabilitation, and fulfillment of their duty to the community. Although the prisoner's work may improve the prison's efficiency and economy, it's not seen as furthering the prison service's business interests.

Court of Appeal decision. The Court of Appeal overturned the previous ruling. It argued that the work done by prisoners in the kitchen was crucial to the prison's operation, and if not performed by prisoners, it would have to be done by someone else. Therefore, the work was performed on behalf of the prison service and for its benefit, as part of its operations and running of the prison. In essence, the prison service gained from this work, so it should also bear its responsibilities. Although the relationship between the prisoners and the prison service was not a typical employment one, as the prisoners were connected to the prison service not by agreement but by their sentences and their wages were nominal, these differences actually made the relationship even closer to an employment one. It was based on obligation rather than mutuality.

Supreme Court decision. The Supreme Court held in favour of the claimant. They found that the defendant, the Ministry of Justice, was vicariously liable

for the prisoner's negligence. This was because the prisoners worked for the defendant's benefit, which created the risk of negligence.

Argumentative analysis

The trial judge held that the Ministry of Justice was not vicariously liable because the relationship between the defendant (Ministry of Justice) and the wrongdoer was not sufficiently relevant. In other words, the judge undermined the second premise of the scheme by answering negatively to the critical question related to vicarious relationship, thus preventing the inferential step from premises to the conclusion.

However, the Court of Appeal and the Supreme Court found that the relationship between the Ministry of Justice and the prisoner was sufficiently close and that the Ministry of Justice gained advantages from prisoner works, and therefore should also bear its responsibilities. In this way the judges allowed the inferential passage from the three premises to the conclusion.

6 Discussion and Limitations

There are two points which are worth discussing regarding the argumentation scheme proposed in this paper. The first point is related to the nature of this scheme and the possibility that this scheme is a more specific implementation of the argument from Rule. The second point is related to whether this schema is applicable universally or not.

Regarding the first point, in the analysed case study, the burden of proof is clearly related to whether or not the vicarious liability should be applied. We showed the most frequent ways in which this schema is supported or attacked (at least in Common Law). When judges argue w.r.t. the applicability of vicarious liability, their arguments mostly focus on Critical Question 1 and Critical Question 2 (related to relationship R and to its "sufficient connection" with wrongdoing W) which can be used to support or undermine (perhaps even undercut) the schema. We also showed where this happens (see critical positions in Figure 1) with specific examples of how different judges undermined the schema (Figure 3). In other words, judges build their arguments in support or attack of such applicability by checking whether some tests is passed (e.g., the "sufficient connection" test). For this reason, we believe that there is a relation with the most general argument from rule. While a simple argument from rule would be too general, this schema can better express the argumentative strategies of judges in the context of vicarious liability. The argument from vicarious liability could therefore be considered a descendant of the argument from rule, but instantiated in the context of liability.

Regarding the second point, although this schema shows a very common argumentative pattern and can be used as it is in the context of Common Law, we believe that the situation might be slightly different in countries which are not under the umbrella of Common Law. In particular, after some first analysis, it seems that we might need more Critical Questions and premises do deal with the legal systems of some countries in Civil Law. For this reason, we think that the

schema proposed in this paper can be considered as a "basic" schema, similarly to how the "basic slippery slope" has been proposed by Walton as the basic pattern underlying more specialised "slippery slope" schemes [13].

Finally, we would like to remark again that this schema should be considered as a first attempt to tackle a huger long-term research goal, namely the analysis of what we call schemes from liability. By shedding some light in this direction, we believe that some interesting discussions can be undertaken. For example, we might understand the way in which argumentative patterns are developed in the context of different kind of liability (secondary liability, strict liability, shared liability, and so on).

7 Conclusion

In this work, we proposed a new direction in the analysis of legal argumentation schemes, focused on the problem of assessing liability in legal argumentation. We started from the assumption that a sufficient account of liability in argumentation schemes is currently missing, despite the pioneering effort in [14]. Furthermore, liability and causality (i.e., legal responsibility and causal responsibility) are topics which are extremely frequent in legal reasoning, but an account of their complex interaction is still missing from the argumentative point of view.

Starting from these two assumptions, we firstly argued that there is need for further exploration of what we call the family of *schemes from liability* and we propose an example of scheme which fits into this category, and which shows a type of interaction between causality and liability, where causal responsibility does not coincide with legal responsibility, as can be seen in many cases of strict liability. We called this scheme Argumentation Scheme from Vicarious Liability.

Furthermore, we showed that our proposed scheme is crucial in the legal tradition, by offering two important and famous case studies where judges discussed whether a vicarious liability was applicable or not by using this kind of scheme.

We then discussed that this schema can be considered a first "basic" schema from vicarious liability, since its applicability outside the Common Law sphere, might require some adjustments. Furthermore, we argued that it might very likely be a descendant of the argument from rule, despite being specifically related to context of vicarious liability.

We are currently working on the development of a computational model for this scheme. In the future, further works are needed to describe the family of argumentation schemes from liability, and their internal relations, from an ontological and logical point of view. Moreover, while we showed a type of interaction between causality and liability, the complexity of their interplay in argumentation deserve further explorations. Finally, the potential relations between these schemes from liability and other existing schemes should be explored more in depth, for example in terms of what kind of schemes can be frequently found in support or attack of these schemes from liability, including the interaction with new compendium of legal schemes proposed in [14].

References

1. Contissa, G., Laukyte, M., Sartor, G., Schebesta, H.: Assessing liability with argumentation maps: An application in aviation law. In: JURIX. pp. 73–76 (2013)
2. Feng, V.W., Hirst, G.: Classifying arguments by scheme. Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies pp. 987–996 (2011)
3. Hahn, U., Bluhm, R., Zenker, F.: Causal argument. Oxford, UK: Oxford University Press (2017)
4. Lawrence, J., Reed, C.: Argument mining using argumentation scheme structures. COMMA pp. 379–390 (2016)
5. Liga, D.: Argumentative evidences classification and argument scheme detection using tree kernels. Proceedings of the 6th Workshop on Argument Mining pp. 92–97 (2019)
6. Liga, D., Palmirani, M.: Argumentation schemes as templates? combining bottom-up and top-down knowledge representation. CMNA@ COMMA pp. 51–56 (2020)
7. Liga, D., Palmirani, M.: Uncertainty in argumentation schemes: Negative consequences and basic slippery slope. CLAR pp. 259–278 (2020)
8. Macagno, F.: Argumentation schemes in ai. Argument and Computation **12**(3), 287–302 (2021)
9. Macagno, F., Walton, D., Reed, C.: Argumentation schemes. history, classifications, and computational applications. History, Classifications, and Computational Applications (December 23, 2017). Macagno, F., Walton, D. & Reed, C pp. 2493–2556 (2017)
10. Moore, M.S.: Causation and responsibility: An essay in law, morals, and metaphysics. Oxford University Press on Demand (2009)
11. Toulmin, S.E.: The uses of argument (1958)
12. Walton, D.: Argument from definition to verbal classification: The case of redefining'planet'to exclude pluto. Informal Logic **28**, 129–154 (2008)
13. Walton, D.: The basic slippery slope argument. Informal Logic **35**(3), 273–311 (2015)
14. Walton, D., Macagno, F., Sartor, G.: Statutory Interpretation: Pragmatics and Argumentation. Cambridge University Press (2021). https://doi.org/10.1017/9781108554572
15. Walton, D., Reed, C., Macagno, F.: Argumentation schemes (2008)
16. Wright, R.W.: Causation in tort law. Calif. L. Rev. **73**, 1735 (1985)

Legitimacy Detection System based on Interpretation Schemes for AI Vehicles Design

Yiwei Lu[1], Zhe Yu[2⋆], Yuhui Lin[3], Burkhard Schafer[1], Andrew Ireland[3], and Lachlan Urquhart[1]

[1] Edinburgh Law School, University of Edinburgh, Edinburgh, UK
[2] Institute of Logic and Cognition, Department of Philosophy, Sun Yat-sen University
[3] School of Mathematical and Computer Sciences, Heriot-Watt University
Y.Lu-104@sms.ed.ac.uk, zheyusep@foxmail.com,
{B.Schafer, lachlan.urquhart}@ed.ac.uk, {y.lin,ceeai}@hw.ac.uk

Abstract. As AI products continue to evolve, the challenges they pose have not only been directed at reforming the legal system, but equally put pressure on the developers of AI products. The ambiguity and uncertainty of the law make it more expensive to test the legality of AI product design solutions. We are interested in developing intelligent support systems that provide legal guidance for products that have an AI component. Here we are concerned with autonomous vehicles where AI plays a significant role. We build upon previous research by using legal ontologies as information carriers and argumentation theory as a reasoning tool. This paper introduces the notion of a legal interpretation scheme – a mechanism that uses context to prioritize the application of abstract legal principles. We argue that this gives a level of flexibility that compensates to some extent for the rigidity of similar legal support systems.

Keywords: Legal ontology · Autonomous vehicle · Legal reasoning · Argumentation theory · Explainable AI.

1 Introduction

The rapid development of autonomous vehicles has created a new set of challenges for manufacturers and engineers in terms of how to make their products compliant with the law. These challenges come from the plethora of new regulations for the new technology, where competing regulations often overlap, or introduce ambiguity or uncertainty of the law itself. As pointed out by van Engers et al. [36], law is based on a dialectical process. This is the consequence of the inevitable introduction of ambiguity, i.e., Hart's notion of "open text", and the inconsistencies that are typically resolved through an adversarial debate. For example, an engineer may want to determine whether

⋆ Corresponding Author.
Work on this paper was supported by Trustworthy Autonomous Systems EP/V026607/1 and AISEC EP/T026952/1As per University of Edinburgh policy, for the purpose of open access, the authors have applied a 'Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

an autonomous driving system should be designed to have the authority to make the decision to run a red light in an emergency. The laws that currently exist for humans are likely to be semantically open-textured or they are not stated explicitly at all in the Road Traffic Act but refer to the background knowledge of judges about the general applicability of exemptions and excuses. Eventually, litigation will clarify some of these terms, but for the designer , it can be difficult to predict if their design choices are lawful. Is an autonomous driving system simply an instance of the legal concept of "the driver" or can the designer ignore rules that obviously do not apply to AVs, such as the excuses that we make for drivers who have to react in split seconds (the "agony of the moment" defence in the common law)? Is the process of making adjustments likely to have unanticipated effects on other laws? For example, could the process of modifying the design to comply with traffic laws behave in a way that violates the laws on private data protection?

The troublesome aspects of this problem for engineers are that: (1) the program design needs to be precisely defined to ensure that the system can make a specific decision in a given situation; (2) this test of the design is complex because a great number of legal principles and rules are involved; (3) such legal service is required with high frequency because designing an intelligent system requires a great deal of adjustments and experimentation, and each design revision requires timely legal feedback; and (4) it is in fact difficult for engineers to guess which direction the law will go for some currently unspecified legal provisions.

Given these reasons, an intelligent system that provides real-time legal support can offer the necessary assistance to engineers and manufacturers of autonomous vehicles. In order to meet the needs of the designers, the system needs adequate legal information and the ability to reason in a timely manner. Indeed, a range of legal information technologies have evolved considerably to meet similar requirements like legal ontology and expert systems. Different logical methods have also been introduced to extend the functionality of intelligent systems that support legal reasoning, including argumentation theory. In previous work [26,25], we introduced ASPIC+, a structured argumentation framework that combines legal ontologies with description logic. ASPIC+ is able to reason about uncertain and inconsistent ontologies. This paper extends this work.

However, it can be foreseen that if such an intelligent system is to be legally applied in the design and production process of autonomous vehicles, it must first pass legal scrutiny. The focus of the scrutiny is not only on whether the reasoning it makes under the same circumstances is consistent with the actual legal outcomes, but also on whether or not its reasoning processes are consistent with the principles of legal reasoning. In other words, this reasoning system not only needs to have logical explainability but also needs to have legal explainability.

Currently existing legal reasoning systems often start by reasoning directly from facts and norms. Whether they support reasoning under priority or not, they often assume that abstract legal principles always have a fixed priority ordering. For example, that the safety of human life always takes priority over property safety. Moreover, this priority order can be mapped to specific legal rules that it generates. However, this design is actually closer to people's common sense of the concept 'reasoning' and differs somewhat from the connotation of legal reasoning. If legal reasoning is understood as

the process of judging legal effects through legal conditions and ultimately arriving at legal solutions, the form is indeed very similar to the reasoning rules triggered by the antecedent. But it actually contains multiple layers of interpretation. To put it simply, legal reasoning involves a process of comparing, reinterpreting, and arguing to determine how existing laws apply to a new case. Therefore, the comparison between legal principles and legal reasoning rules is not a matter of common sense priority but of who is more appropriate in a specific legal context.

There is a suitable example that can illustrate the difference between the legal reasoning process in reality and the reasoning systems when facing new concepts. When property law and privacy law were established, GPS had not yet been invented. Therefore, neither the definition of property nor the definition of privacy included GPS and its location information. So, did the police have the right to investigate through GPS location information? In 2012, the US Supreme Court ruled that tracking GPS without a search warrant is illegal. Justice Scalia stated in the court's opinion that GPS tracking violated others' property, making it illegal. Justice Sotomayor, in the concurring opinion of Justice, interpreted this behavior as an infringement of privacy. Although both reached the same conclusion, the applicable legal interpretation was different, which could have a significant impact on the results of similar cases.

From the perspective of the designer, sometimes these different explanations don't matter – it is only the behaviour of the car that counts after all, not which legal norms it tries to comply with when doing so. But sometimes the different justifications can yield different design decisions. In the above example, the outcome of the decision could mean to integrate the GPS unit in such a way that it does not disclose information from a remote request by the police unless a proper authorization (the electronic equivalent of the warrant) is transmitted too. But under the property solution, it makes it safer also to share the information with third parties. The data protection rationale makes this more risky, as under US law, once data has been disclosed to third parties, the privacy interest is relinquished. So if the designer wants to maximize both the privacy of the owner and the comfort that comes with using apps that rely on GPS, they may reach different solutions depending on whether they follow Scalia or Sotomayor: The former allows more data transfer also to app providers, the second prefers edge computing and keeping the data under the control of the driver.

Currently, most illegal access to GPS data is interpreted as an infringement of privacy rather than property, indicating that the legal context in most cases is closer to the connotation of the right to privacy. This is a trade-off in terms of applicability, rather than a permanent preference for the right to privacy over property rights. Current legal reasoning systems can hardly demonstrate the process of choosing legal interpretations in different contexts, which can lead to problems. For example, if we set the priority to be the driver's safety rather than the safety of property, and directly interpret the legal implications of the autonomous driving system based on this principle without discussing the legal context, it will make the car unhesitatingly choose to sacrifice the owner's interests like important goods to protect the car as a driver.

Therefore, inspired by Amgoud and Parsons [3], this paper extends the legal support system *LeSAC* developed in the previous study [26,25]. We change the original model of reasoning directly from the facts and norms to include a process of selecting a legal

interpretation scheme before proceeding with the reasoning. In other words, we deepen the original single-layer structure into a double-layer structure, assigning the priority order of legal principles based on the selection of legal interpretation schemes, and ultimately bringing the priority order of the generated legal arguments. In this way, the applicable legal principles, rules and their priority are determined by a legal interpretation, rather than by applying common elements to all cases. Therefore we eliminate the damage caused by the original method to the legal reasoning process, and increase the explainability in terms of law. For engineers using this system, making corresponding legal decisions by different legal contexts and corresponding interpretation can greatly increase the legality of the product. The construction of this system is a gradual process involving several studies and related papers. This paper focuses on showing how the system accomplishes legal reasoning and gives explanations when legal interpretation schemes are already given.

In the following, we will first discuss some related works and further clarify the significance of this article through comparison. Then, we will demonstrate the structure of the new legal reasoning framework through a case study and a series of formal definitions. Next, we will use this case study to demonstrate how the framework can answer various consultations needed by engineers during the design process of autonomous vehicles through reasoning. Finally, we will summarize this paper and provide a perspective on future research directions.

2 Related Work

Road rules are more detailed and precise than "top level" legal regimes such as the general law of delict. They are also a good candidate for representation in a logic framework [28]. So we choose road rules as a starting point. Considering the power to contain enough information, legal ontology, a classical representation format, has been used as the basic legal information carrier. There are works capturing legal knowledge and reasoning on the abstract level: the Legal Knowledge Interchange Format (LKIF) Core ontology builded on the Web Ontology Language (OWL) and LKIF rules [23,2]; the Core Legal Ontology (CLO) based on the extension of the DOLCE (DOLCE+) foundational ontology [18]; the LRI-Core ontology aimed at the legal domain grounded in commonsense [10], UOL[22] and the Functional Ontology for Law (FOLaw) [34,35]. And there are also works for specified legal domains and their reasoning methods: Ukraine legal ontology[20], MCO[30] and JUR-IWN[11]. But the mainstream language of the current legal ontology is the Web Ontology Language (OWL) or OWL2, whose semantics are based on DL [4], a subset of first-order logic. This determines that the reasoning in legal ontologies has to be under predefined and fixed priority (all equal) and consistent context, which seriously restrict the reasoning function and its reflection of real legal world. That is also why current legal ontologies mainly play the role as documents or concepts managers. Detecting and repairing inconsistent parts [32,17] or extending classical logic by adding true values [38] are possible ways to extend the functions but they weaken the reasoning strength of DL [38] and require frequent expert guidance[32,17]. And that's why formal argumentation could play a role in solving this problem.

In [6], defeasible logic has been imported for traffic rules compliance checking of automated vehicle maneuvers. It has shown a great performance in interpreting legal

concepts and applying rules to actions like overtaking other cars. To reflect more on the interactions between legal arguments, this paper uses formal argumentation theory. From [14], it has been shown that formal argumentation is a strong tool for reasoning under inconsistent and uncertain contexts [15,19,21,29]. In [21,27], DL ontology is expressed as Defeasible Logic Programs (*DeLP*) [19], while paper [9] present argumentation frameworks based on the Deductive Argumentation [5] framework. But these are not argumentation frameworks especially designed for legal application. So when applied to legal problems, technical issues will arise. For example, they lack diverse description of relationships among arguments for complicated interactions between legal claims; they have no mechanism to resolve situations where two legal principles need to be compared and; the formal explanation of reasoning results is not given. Based on these reasons, we chose to extend *ASPIC*$^+$ [31] to handle the legal reasoning in previous works [26,25], which is originally constructed for legal application. Although it offers many advantages like powerful description of relationships, reflection of an agent's attitudes and transparent reasoning processes, it still has space to be extended. In our previous work, we applied legal principles using fixed priorities. Relaxing this fixed priority approach is the core contribution of this paper.

Borning et al. [8] introduce a constraint hierarchy that involves a collection of constraints, each designated as either essential or preferred at varying levels of importance. The theory of constraint hierarchies explores different methods for choosing among multiple potential solutions and establishes several theoretical relationships between these options. In comparison, we resolve conflicts between arguments by priorities legal principles associated with norms, and the final selection of explanation schemes will be based on whether the combination can provide a satisfactory legal interpretation.

Further more, as discussed above, explanability is important to building trust in the legal advice provided. Not only it is the core nature of how law works in real life, but also it has abyssal impacts on similar cases for future and standards for AI product design. Argumentation theory strongly supports explanations for different situations. In [13], a comprehensive literature survey of explainable AI research is provided from an argumentation perspective. In [7], a flexible framework that provides explanations about why a claim is finally accepted or rejected is described, which uses various extension-based semantics in argumentation frameworks. However, current explanations based on argumentation theories still mainly focus on explaining why an argument is accepted or not according to specific relationships within arguments [16]. These correlations may include attack, support, preferred or others. But in legal cases, why the relationships should be like this matters a lot, that is, how to select a proper legal interpretation before doing norm-based reasoning matters. This is another reason why we extend the previous framework in this paper to give the reasoning results more convincing legal explanation.

3 Legal ontology and argumentation & a case study

To help clarify the principles of this system, we use a case. Encoding of this case study is available in [1].

Consider the following scenario that engineers may face if their task is to design AV that behaves in a law-compliant way:

Example 1. Currently, the law stipulates a number of behaviours that a human driver has to observe after an accident has happened. This includes a duty to stay at the scene of an accident and to provide first aid if necessary and feasible. The design question now is, does a driverless car "inherit" this obligation in the same way it "inherits" from the human driver the duty to stop at a traffic light? A recent proposal by the Scottish and English Law Commissions differentiates between functions that are core to the safe operation of an AV and those that are auxiliary.

As a complication, let's assume there is one passenger in the car, but he is (illegally) too drunk to do anything. In such a "contrary to duty" scenario, how should the AV car react now when somebody is hit? Should it just report to the police and keep doing its original job: Sending the passenger to destination as soon as possible? If the injured party is likely to die if not receiving medical aid in time, is it a legal requirement that the AV stops the only passenger from leaving and asks him to take the responsibility as a driver to give some help? Especially considering the passenger is drunk, what if he does second harm to the injury or put himself in danger? Here road traffic law interacts with other, more general legal provisions about duty of care. So what kind of legal interpretation will be applied will make a huge difference.

To handle this possible situation, we refer to current and relevant legal rules. We extract and select some most relevant information from traffic law and criminal law:

(1) It is illegal to drive a motor vehicle while intoxicated. People who drive while intoxicated shall lose their driving license and may be prosecuted in criminal law.

(2) A person who commits a hit-and-run accident will be criminal responsibility, especially when the escape causes the death or the driver is intoxicated.

(3) When an accident happens, the driver should take the responsibility to transfer the injured party to a safe place and provide aid if the situation is urgent.

Obviously, the concept of a driver in the law for humans is based on common sense and the application of autonomous vehicles changes the certainty of this concept. If we are unable to decide who applies to the concept of driver in such a situation under the established law of self-driving systems versus passengers, the system will not be able to react in any way in an emergency situation. And different legal interpretations can lead to different results and problems. If the automated driving system is interpreted as the driver and not the passenger, this means that the passenger does not bear any liability. Therefore, in order to protect the passenger, it should not stop and let the passenger out of the car. However, if the autonomous vehicle assumes liability, it should provide physical assistance, which is not possible. If an intoxicated passenger is interpreted as the driver, it would seem that the autonomous vehicle should ask him to get out of the car and provide assistance. But how should the duty to protect the passenger be handled in that case? If the passenger is injured, how should liability be determined? On the basis of the above analysis we can see that the criteria and conclusions of the reasoning are, in fact, determined by differences in the interpretation of the law.

If we assume it should be the AI car's job to make sure no more risk will occur to the drunk passenger, which is highly possible. It means encouraging the passenger to get the car off on the road could be against the law. To ad this into the consideration, we import one more legal rule into this example:

(4) AI vehicles must avoid letting their drunk passengers get off the car in the middle of the route.

3.1 Legal Ontology

As mentioned above, DL is the basic semantic of OWL or OWL2, which are the main logic languages of current legal ontologies. DLs are a family of knowledge representation formalisms. The basic notions of DL systems are *concepts* and *roles*. A DL system contains two disjoint parts: the TBox and the ABox. TBox introduces the terminology, while ABox contains facts about individuals in the application domain. There are many DLs and the legal ontology in this paper is built upon the *ALC* expression [33,4].

In a legal ontology, legally reasoning rules such as traffic rules will be allocated into Tbox, while explicit legal designs, e.g. AVs will stop or not, will be in ABox. As for the situation in **Example** 1, a legal ontology's TBox will be:

Example 2 (DL encoding of Example 1, TBox).

$$T = \left\{ \begin{array}{l} Driver \sqsubseteq Sober;\ Sober \sqcap Intoxicated \sqsubseteq \emptyset;\ Intoxicated \sqcap LeaveCar \sqsubseteq \emptyset; \\ Driver \sqcap Intoxicated \sqsubseteq BeRevokedDrivingLicense \sqcap TakeCriminalResposibility; \\ \exists hitAndRun.Injury \sqsubseteq TakeCriminalResposibility; \\ \exists hitAndRun.causeDeath \sqsubseteq AggravatedPunishment; \\ \exists hitAndRun.Injury \sqcap Driver \sqcap Intoxicated \sqsubseteq AggravatingPunishment; \\ CauseAccident \sqcap Injury \sqsubseteq \exists transferToSafePlace.Injury; \\ CauseAccident \sqcap NeedEmergencyAid.Injury \sqsubseteq doNecessaryAid; \\ (transferToSafePlace \sqcup doNecessaryAid) \sqcap \neg LeaveCar \sqsubseteq \emptyset \end{array} \right\}$$

Assuming the AV system named "AC1", we consider the passenger should take the responsibility as the current legal concept "driver". When an AV hits somebody named "Injury1" on the road, it will ask the only passenger named "PS1" to leave the car and help the injured party, no matter if he is drunk or not. The corresponding ABox is:

Example (Example 2 cont. DL encoding of ABox)**.**

$$A = \left\{ \begin{array}{l} Driver(PS1);\ Intoxicated(PS1);\ hitAndRun(PS1, Injury1);\ Injury(Injury1); \\ causeDeath(PS1, Injury1);\ CauseAccident(PS1);\ NeedEmergencyAid(Injury1) \end{array} \right\}$$

3.2 L-ASPIC

In previous papers [26,25] we built our theory on an argumentation system for legal reasoning [37], called *L-ASPIC* , which is a simplified version of *ASPIC*$^+$ framework proposed in [31]. In the current paper, we retain the basic design ideas of *L-ASPIC* and provide the following definitions based on the new design.

Definition 1 (Argumentation system). *An L-ASPIC argumentation system (L-AS) is a tuple* $(\mathscr{L}, \mathscr{R}, n)$*, where*

- $\mathscr{L}$ *is a set of formal language closed under negation* ($\neg$)*, where* $\psi = -\varphi$ *means* $\psi = \neg\varphi$ *or* $\varphi = \neg\psi$*;*

- $\mathscr{R} = \mathscr{R}_s \cup \mathscr{N}$ *is a set of strict inference rules* ($\mathscr{R}_s$) *of the form* $\varphi_1, \ldots, \varphi_n \rightarrow \varphi$, *and legal norms* ($\mathscr{N}$) *based on defeasible inference rules, of the form* $\varphi_1, \ldots, \varphi_n \Rightarrow \varphi$ ($\varphi_i, \varphi \in \mathscr{L}$); *let* $\mathscr{R}_s \cap \mathscr{N} = \emptyset$;
- n *is a naming function such that* $n : \mathscr{N} \rightarrow \mathscr{L}$.

Let $\Delta = (T, A)$ be a legal ontology for AV based on description logic, given an *L-AS*, (*L-AS*, $\mathscr{K}^A$) is an argumentation theory about Δ (denoted as *L-AT*$^\Delta$), where *L-AS* = ($\mathscr{L}, \mathscr{R}^T, n$), such that $\mathscr{R}^T$ is the set of rules corresponding to T (a mapping table can be found in [24,37]), and $\mathscr{K}^A$ is the set of premises based on A.

Let all the formulas in $\mathscr{K}$ that are used to build an argument be denoted as $\texttt{Prem}$, all its sub-arguments be denoted as $\texttt{Sub}$, all the applied rules be denoted as $\texttt{Rules}$, and the consequent of the last rule be denoted as $\texttt{Conc}$. Arguments constructed based on *L-AT*$^\Delta$ are defined as follows.

Definition 2 (Argument). *An argument* α *constructed based on L-AT*$^\Delta$ *has one of the following forms:*

1. φ, *if* $\varphi \in \mathscr{K}^A$, *such that* $\texttt{Prem}(\alpha) = \{\varphi\}$, $\texttt{Conc}(\alpha) = \varphi$, $\texttt{Sub}(\alpha) = \{\varphi\}$, *and* $\texttt{Rules}(\alpha) = \emptyset$;
2. $\alpha_1, \ldots, \alpha_n \rightarrow \psi$ *if* $\alpha_1, \ldots, \alpha_n$ *are arguments, such that there exists a rule* $\texttt{Conc}(\alpha_1), \ldots, \texttt{Conc}(\alpha_n) \rightarrow \psi$ *in* $\mathscr{R}^T$, *and* $\texttt{Prem}(\alpha) = \texttt{Prem}(\alpha_1) \cup \ldots \cup \texttt{Prem}(\alpha_n)$, $\texttt{Conc}(\alpha) = \psi$, $\texttt{Sub}(\alpha) = \texttt{Sub}(\alpha_1) \cup \ldots \cup \texttt{Sub}(\alpha_n) \cup \{\alpha\}$, $\texttt{Rules}(\alpha) = \texttt{Rules}(\alpha_1) \cup \ldots \cup \texttt{Rules}(\alpha_n) \cup \{\texttt{Conc}(\alpha_1), \ldots, \texttt{Conc}(\alpha_n) \rightarrow \psi\}$;
3. *Replace all the "*$\rightarrow$*" in point 2 by "*$\Rightarrow$*".*

The conflicts (attack relation) between arguments is defined as follows.

Definition 3 (Attacks). *Let* α, β, β' *be arguments constructed based on an L-AT*$^\Delta$ = (*L-AS*, $\mathscr{K}^A$), α *attacks* β *on* β', *iff: 1)* $\beta' \in \texttt{Sub}(\beta)$ *of the form* $\beta''_1, \ldots, \beta''_n \Rightarrow \varphi$ *and* $\texttt{Conc}(\alpha) = -\varphi$; *or 2)* $\beta' = \varphi$ *and* $\varphi \in \texttt{Prem}(\beta) \cap \mathscr{K}$, *s.t.* $\texttt{Conc}(\alpha) = -\varphi$.

When two arguments are in conflict, whether one can defeat another should be determined by some pre-defined priorities (these could e.g. be higher-order legal values depending on the legal context such as "respect for human life"). In the current paper, such priorities are reflected by $\leqslant$ in the legal interpretation scheme.

First we defined scheme modules, which can be combined to become a scheme for explanation. Every scheme module is composed of two elements: a set of legal principles and the priority orderings on them. This means a set of legal interpretation principles when a certain situation happened. For example, when real human's life is in danger, lives are more important than property. Or, when not commercial but personal data is leaked, privacy law is more applied than property law. In a case, multiple scheme moudles may be applied as a whole, following the next definitions:

Definition 4 (Scheme Module). *A scheme module* S_i *is a tuple* $(P_i, \leqslant_i)$, *where* $P_i = \{p_x, p_y, \ldots\}$ *is a finite set of legal principles and* $\leqslant_i$ *is an ordering on* P_i.

For any p_x and p_y, we write $p_x <_i p_y$ if and only if $p_x \leqslant_i p_y$ and $p_y \nleqslant_i p_x$, and $p_x =_i p_y$ if and only if $p_x \leqslant_i p_y$ and $p_y \leqslant_i p_x$. And we assume that addition can be performed between the orderings, defined as follows.

Definition 5 (Priority addition). *For any S_i and S_j, if $\nexists p_x$, $p_y \in P_i \cup P_j$ such that $p_x <_i p_y$ and $p_y <_j p_x$, then let $\leqslant = \leqslant_i + \leqslant_j$, such that:*

1. *$p_x \leqslant p_y$, iff $p_x \leqslant_i p_y$ or $p_x \leqslant_j p_y$;*
2. *$p_x < p_y$, iff $p_x \leqslant p_y$ and $p_x \not\leqslant_i p_y$ or $p_x \not\leqslant_j p_y$;*
3. *$p_x = p_y$, iff $p_x \leqslant p_y$ and $p_y \leqslant p_x$*

L-ASPIC assumes that each legal norm is associated with a primary legal principle in $\mathscr{P}$ (whereas one legal principle may be associated with multiple norms), so an interpretation scheme should contain legal principles for all the norms in $\mathscr{N}$, as well as the mapping from principles to norms. Therefore, we define an interpretation scheme composed by scheme modules as follows.

Definition 6 (Scheme). *Let $S_i, \ldots, S_k$ be scheme modules and $\mathscr{N}$ the set of legal norms of L-AT$^\Delta$. $\mathscr{S} = (\mathscr{P}, \leqslant, prin)$ is a scheme for legal interpretation, such that $\mathscr{P} = P_i \cup \ldots \cup P_k$, $\leqslant = \leqslant_i + \ldots + \leqslant_k$ is an ordering on $\mathscr{P}$ according to $\leqslant_i, \ldots, \leqslant_k$, and prin is a total function such that $prin : \mathscr{N} \to P$.*

By Def. 5 and Def. 6, we may get more than one qualified schemes. The comparison of these schemes will be discussed in future work.

For any argument α constructed based on an *L-AT*$^\Delta$, let $\texttt{LastNorms}(\alpha) = \emptyset$ if $\texttt{Rules}(\alpha) \cap \mathscr{N} = \emptyset$, or $\texttt{LastNorms}(\alpha) = \{\texttt{Conc}(\alpha_1), \ldots, \texttt{Conc}(\alpha_n) \Rightarrow \psi\}$ if $\alpha = \alpha_1, \ldots, \alpha_n \Rightarrow \psi$, otherwise $\texttt{LastNorms}(\alpha) = \texttt{LastNorms}(\alpha_1) \cup \ldots \cup \texttt{LastNorms}(\alpha_n)$. The set $\{prin(N) | N \in \texttt{LastNorms}(\alpha)\}$ is denoted as $\texttt{LastPrin}(\alpha)$.

Let $\mathscr{A}$ denote all the arguments constructed based on an *L-AT*$^\Delta$ and $\trianglelefteq_{Dem}$ denote a set comparison based on the *Democratic* approach [12]. The preference ordering $\preceq$ on $\mathscr{A}$ is defined as follows.

Definition 7 (Argument ordering). *Let $(L\text{-}AS, \mathscr{K}^A)$ be an L-AT$^\Delta$, for all α, β constructed based on it, $\beta \preceq \alpha$ iff $\texttt{LastPrin}(\beta) \trianglelefteq_{Dem} \texttt{LastPrin}(\alpha)$, i.e.: 1) $\texttt{LastPrin}(\alpha) = \emptyset$ and $\texttt{LastPrin}(\beta) \neq \emptyset$; or 2) $\forall p_x \in \texttt{LastPrin}(\beta)$, $\exists P_y \in \texttt{LastPrin}(\alpha)$ s.t. $p_x \leqslant p_y$.*

For the choice of comparative principles, preferences on the set of arguments are extracted according to the *Democratic* approach for set comparison [12] and the *last-link principle* [31] for the elements selection.

In order to get an output of acceptable conclusions, we need to identify the justified arguments, which can be achieved by an argument evaluation process based on abstract argumentation frameworks (*AF*) and argumentation semantics [14]. Given an *L-AT*$^\Delta$, an $AF = (\mathscr{A}, \mathscr{D})$ can be established based on the set of all the arguments ($\mathscr{A}$) and the defeat relation ($\mathscr{D}$) on the basis of the attack relation between arguments and the ordering $\preceq$ on $\mathscr{A}$. Let S denote one of the basic argumentation semantics introduced in [14], $\mathscr{E}_S$ denote the set of all extensions obtained under S, and $E_S \in \mathscr{E}_S$ denote one of the extensions. An argument $\alpha \in \mathscr{A}$ is said to be *accepted* w.r.t. E_S if $\alpha \in E_S$. In the following we say α is *sceptically justified* under S if $\forall E_S \in \mathscr{E}_S$, $\alpha \in E_S$, and α is *credulously justified* under S if $\exists E_S \in \mathscr{E}_S$ such that $\alpha \in E_S$. Then according to the accepted/justified arguments, we can identify the accepted conclusions.

4 Legal support system for autonomous vehicles

In §3.2 we defined an argumentation framework for reasoning based on an inconsistent legal ontology with an interpretation scheme. Then given a legal ontology, particularly for AVs design, we can construct a *LeSAC* system:

Example 3 (A LeSAC). Given a legal ontology for AV $\Delta = (T, A)$, as shown in Example 2. $LeSAC = (L\text{-}AT^{\Delta}, \mathscr{S})$ is an argumentation theory instantiated by Δ:

$$\mathscr{N} = \left\{ \begin{array}{l} r_1 : Driver(x) \Rightarrow Sober(x); \\ r_2 : Intoxicated(x) \Rightarrow \neg LeaveCar(x); \\ r_3 : Driver(x), Intoxicated(x) \Rightarrow BeRevokedDrivingLicense(x); \\ r_4 : Driver(x), Intoxicated(x) \Rightarrow TakeCriminalResposibility(x); \\ r_5 : hitAndRun(x,y) \Rightarrow TakeCriminalResposibility(x); \\ r_6 : hitAndRun(x,y), causeDeath(x,y) \Rightarrow AggravatedPunishment(x); \\ r_7 : hitAndRun(x,y), Driver(x), Intoxicated(x) \Rightarrow AggravatedPunishment(x); \\ r_8 : CauseAccident(x), Injury(y) \Rightarrow transferToSafePlace(x,y); \\ r_9 : CauseAccident(x), Injury(y), NeedEmergencyAid(y) \Rightarrow doNecessaryAid(x,y) \end{array} \right\}$$

$$\mathscr{R}_s = \left\{ \begin{array}{l} r_{10} : Sober(x) \rightarrow \neg Intoxicated(x); \\ r'_{10} : Intoxicated(x) \rightarrow \neg Sober(x); \\ r_{11} : transferToSafePlace(x,y) \rightarrow LeaveCar(x); \\ r'_{11} : \neg LeaveCar(x) \rightarrow \neg transferToSafePlace(x,y); \\ r_{12} : doNecessaryAid(x,y) \rightarrow LeaveCar(x); \\ r'_{12} : \neg LeaveCar(x) \rightarrow \neg doNecessaryAid(x,y) \end{array} \right\}$$

$$\mathscr{K}^A = \left\{ \begin{array}{l} Driver(PS1); Intoxicated(PS1); \\ hitAndRun(PS1, Injury1); \\ Injury(Injury1); \\ causeDeath(PS1, Injury1); \\ CauseAccident(PS1); \\ NeedEmergencyAid(Injury1) \end{array} \right\}$$

$$\mathscr{P} = \left\{ \begin{array}{l} p_1 : Prompt\ measures\ should\ be\ taken\ to\ protect\ lives\ when\ causing\ injury\ to\ others; \\ p_2 : AI\ products\ must\ avoid\ physical\ risk\ from\ their\ behaviors\ for\ their\ users; \\ p_3 : People\ should\ avoid\ putting\ others\ into\ dangerous\ by\ his\ own\ behaviours, \\ \quad and\ should\ bear\ corresponding\ responsibility. \end{array} \right\}$$

Suppose there are two qualified schemes $\mathscr{S}$ and $\mathscr{S}'$. $\mathscr{S}$ is based on the idea of considering the intoxicated passenger as the driver. Due to the severity of the victim's injuries and the driver's fault, as well as his ability to take action, p_1 takes priority over p_2, i.e. $p_2 < p_1$.

$\mathscr{S}'$ is based on the idea of considering the AV system as the driver. Due to the faultlessness of the passenger, he should not bear additional risks. p_2 takes priority over p_1, i.e. $p_1 < p_2$.

The sets of principles and the mappings from norms to principles are the same for $\mathscr{S}$ and $\mathscr{S}'$:

$prin(r_1) = p_3$; $prin(r_2) = p_2$; $prin(r_3) = p_3$; $prin(r_4) = p_3$; $prin(r_5) = p_3$; $prin(r_6) = p_3$; $prin(r_7) = p_3$; $prin(r_8) = p_1$; $prin(r_9) = p_1$

Properties for a well-defined argumentation theory based on $ASPIC^+$ are specified in [31]. Likewise, a well defined *LeSAC* should also meet some requirements, such

as $\mathscr{R}_s$ should be closed under transposition or contraposition. closure under transposition (or contraposition). ie.: if $\varphi_1,\ldots,\varphi_n \to \psi \in \mathscr{R}_s$, then for each $i = 1\ldots n$, there is $\varphi_1,\ldots,\varphi_{i-1},-\psi,\varphi_{i+1},\ldots\varphi_n \to -\varphi_i \in \mathscr{R}_s$. In **Example** 3, rules r'_{10}, r'_{11} and r'_{12} are the transposed rules of rule r_{10}, r_{11} and r_{12}, respectively.

We now present *LeSAC* its reasoning functions using the case study. To clarify these support functions more visually, we use Example 3 to show how AV designers could solve their different problems in this situation through *LeSAC*.

Legal compliance detection When engineers complete a whole design draft, they could use the consistency checking function to check whether this design is fully compliant with given laws and where conflicts are under certain interpretation.

Definition 8 (Consistency Checking).
The ABox of Δ is consistent w.r.t. the TBox of Δ iff $\mathscr{A}$ is conflict-free based on attack relations, i.e., $\nexists \alpha,\beta \in \mathscr{A}$ such that α attacks β.

If a design is completely consistent after reasoning, it means it is fully compliant with given laws. Otherwise, it is not. And by tracing where arguments conflict, we could know which part of the design needs modification. Based on the *LeSAC* in Example 3, we can at least construct the following two arguments.

Example (Example 3 cont.)**.**
$\alpha = (CauseAccident(PS1), Injury(Injury1) \Rightarrow transferToSafePlace(PS1, Injury1)) \to LeaveCar(PS1)$ *and* $\beta = Intoxicated(PS1) \Rightarrow \neg LeaveCar(PS1)$. *According to Definition 3, α and β attack each other, therefore the legal ontology on which this LeSAC is based is inconsistent.*

Feedback for single change If AV engineers want to keep the main design of an AV and only do some minimal changes, *LeSAC* can provide possible further legal consequences with these new details by instance checking. According to *LeSAC*, assertions are the conclusions of arguments. So based on the extension of arguments, we can decide whether an assertion is accepted. The definition of acceptance of assertions is:

Definition 9 (Assertion Acceptance). *An assertion X is sceptically/credulously accepted under certain argumentation semantics S, iff $\exists A \in \mathscr{A}$, s.t. A is sceptically/credulously justified w.r.t. $\mathscr{E}_S$ and* $\texttt{Conc}(A) = X$.

To determine whether a certain modification is consistent with the current design and given laws, we translate this problem into whether a legal assertion about this AV can be accepted as a conclusion of an accepted/justified argument. Consider arguments α and β in Example 4, we have $\texttt{LastNorms}(\alpha) = \{r_8\}$, $\texttt{LastNorms}(\beta) = \{r_2\}$, and $\texttt{LastPrin}(\alpha) = p_1$, $\texttt{LastPrin}(\beta) = p_2$ respectively. Assume that based on $\leqslant$ on $\mathscr{P}$, $p_2 < p_1$, then according to Definition 7, $\beta \prec \alpha$. Therefore, α can defeat β, but not vice versa. Based on the *LeSAC* in Example 3, there are no other arguments to attack or defeat α. As a consequence, α is sceptically justified w.r.t. any $\mathscr{E}_S$, and the assertion "$LeaveCar(PS1)$" is sceptically accepted. The following definition defines instance checking based on a *LeSAC* for all the possible forms of classes.

Definition 10 (Instances Checking). *Let φ be an individual, sceptically or credulously, it holds that φ is an instance of a class:*

- C ($/\neg C$), *iff* $\exists \alpha \in \mathscr{A}$, *s.t.* α *is sceptically justified w.r.t.* $\mathscr{E}_S$ *and* $\texttt{Conc}(A) = C(\varphi)(/\neg C(\varphi))$;
- $C \sqcap D$, *iff* $\exists \alpha, \beta \in \mathscr{A}$ *such that* α *and* β *are both sceptically/credulously justified w.r.t.* $\mathscr{E}_S$ *and* $\texttt{Conc}(A) = C(\varphi)$, $\texttt{Conc}(B) = D(\varphi)$;
- $C \sqcup D$, *iff* $\exists \alpha, \beta \in \mathscr{A}$ *s.t. at least one of* α *and* β *are sceptically/credulously justified w.r.t.* $\mathscr{E}_S$ *and* $\texttt{Conc}(\alpha) = C(\varphi)$, $\texttt{Conc}(\beta) = D(\varphi)$;
- $\exists P.D$, *iff* $\exists \alpha, \beta \in \mathscr{A}$ *such that* α *and* β *are both sceptically/credulously justified w.r.t.* $\mathscr{E}_S$ *and* $\texttt{Conc}(\alpha) = P(\varphi, x)$, $\texttt{Conc}(\beta) = D(x)$ *(x is an individual);*
- $\forall P.D$, *iff* $\exists \alpha \in \mathscr{A}$ *s.t.* $\texttt{Conc}(\alpha) = P(\varphi, x)$*; and* $\forall \alpha \in \{\alpha | \ \texttt{Conc}(\alpha) = P(\varphi, x)\}$, $\exists \beta \in \alpha$, *s.t.* $\texttt{Conc}(\beta) = D(x)$.

Giving legal explanations How to best use argumentation theory to generate understandable explanations has become an increasingly important topic in AI regulation and AI design. As discussed above, in this paper, an explanation of a legal reasoning should be under a selected legal interpreatation. Based on the selected interpretation scheme, we propose the following formal definition of an explanation:

Definition 11 (Explanation). *Let X be an assertion in a LeSAC that is sceptically or credulously accepted under certain argumentation semantics, then $\exists \alpha \in \mathscr{A}$ s.t. $\texttt{Conc}(\alpha) = X$. The explanation for accepting X is $Exp = \{\leqslant\} \cup \mathscr{C}(\alpha) \cup \mathscr{C}(\beta)$, where:*

- $\leqslant$ *is the priority ordering in selected scheme $\mathscr{S}$;*
- $\mathscr{C}(\alpha) = \texttt{Prem}(\alpha) \cup \texttt{Rules}(\alpha)$*, which explains how X is reached;*
- $\mathscr{C}(\beta) = \texttt{Prem}(\beta) \cup \texttt{Rules}(\beta)$ *such that β defends α according to $\leqslant$ and the defeat relation $\mathscr{D}$, which explains why X is justified.*

Def. 11 provides a formal explanation of why a legal conclusion X is accepted for certain design requirement. It consists of three parts. The first part is based on the legal interpretation scheme defined in Def.6. The second part explains how X is reached by presenting all the premises contained in $\mathscr{K}^A$ and all the legal rules contained in $\mathscr{R}^T$ that are applied to construct argument α, while the third part explains why this legal conclusion is accepted by presenting all the legal information and relevant legal principles applied to construct the arguments that defend α. Consider our running examples, for the acceptance of the assertion "$LeaveCar(PS1)$",

$$Exp = \{p_2 < p_1\} \cup \{Injury(Injury1), CauseAccident(PS1), NeedEmergencyAid(Injury1)\} \cup \{r_8, r_9\}$$

and for the acceptance assertion "$\neg LeaveCar(PS1)$", it is:

$$Exp' = \{p_1 < p_2\} \cup \{Intoxicated(PS1)\} \cup \{r_2\} \cup \{r'_{10}\}$$

5 Conclusion and future work

This paper constructed a legal support system that aims to to help engineers of AVs improve the legal compliance of their designs. We introduced the notion of legal interpretation concept and capture it by adding corresponding elements in our system. In current *LeSAC* , priority ordering for legal principles is given by the selection of legal interpretation schemes, meaning the process of deciding how to apply existing law to new cases. Through a worked example, we have shown how our extension gives rise to a level of reasoning that is closer to real-world applications of law. And thus reasoning explanations are more convincing. In future work, we will mainly focus on two problems: (1) how to extract the legal interpretation automatically; (2) how to compare different interpretations with formal standards.

References

1. JURISIN 23 paper resource webpage. https://colab.research.google.com/drive/1n0QEJVWSiurCu0Rzn4bdkVge3ycC0dHD?usp=sharing, accessed: 2023-04-17
2. Alexander, B.: LKIF core: Principled ontology development for the legal domain. Law, ontologies and the semantic web: channelling the legal information flood **188**, 21 (2009)
3. Amgoud, L., Parsons, S.: Agent dialogues with conflicting preferences. In: Intelligent Agents VIII: Agent Theories, Architectures, and Languages 8th International Workshop, ATAL 2001 Seattle, WA, USA, August 1–3, 2001 Revised Papers 8. pp. 190–205. Springer (2002)
4. Baader, F., Calvanese, D., McGuinness, D., Patel-Schneider, P., Nardi, D., et al.: The description logic handbook: Theory, implementation and applications (2003)
5. Besnard, P., Hunter, A.: Constructing argument graphs with deductive arguments: a tutorial. Argument & Computation **5**(1), 5–30 (2014)
6. Bhuiyan, H., Governatori, G., Bond, A., Rakotonirainy, A.: Traffic rules compliance checking of automated vehicle maneuvers. Artificial Intelligence and Law pp. 1–56 (2023)
7. Borg, A., Bex, F.: A basic framework for explanations in argumentation. IEEE Intelligent Systems **36**(2), 25–35 (2021)
8. Borning, A., Freeman-Benson, B., Wilson, M.: Constraint hierarchies. LISP and symbolic computation **5**, 223–270 (1992)
9. Bouzeghoub, A., Jabbour, S., Ma, Y., Raddaoui, B.: Handling conflicts in uncertain ontologies using deductive argumentation. In: WI'17. pp. 65–72 (2017)
10. Breuker, J., Valente, A., Winkels, R.: Use and reuse of legal ontologies in knowledge engineering and information management. In: Law and the Semantic Web, pp. 36–64 (2005)
11. Casellas, N., Blázquez, M., Kiryakov, A., Casanovas, P., Poblet, M., Benjamins, R.: Opjk into proton: Legal domain ontology integration into an upper-level ontology. In: OTM 2005. pp. 846–855
12. Cayrol, C., Royer, V., Saurel, C.: Management of preferences in assumption-based reasoning. In: IPMU '92. pp. 13–22 (1992)
13. Čyras, K., Rago, A., Albini, E., Baroni, P., Toni, F.: Argumentative xai: A survey. In: Proceedings of IJCAI-21. pp. 4392–4399 (2021)
14. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**(2), 321 – 357 (1995)
15. Dung, P.M., Kowalski, R.A., Toni, F.: Assumption-based argumentation. In: Simari, G., Rahwan, I. (eds.) Argumentation in Artificial Intelligence. pp. 100–218. Springer US (2009)
16. Fan, X., Toni, F.: On computing explanations in argumentation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 29 (2015)
17. Fang, J., Huang, Z.: Reasoning with inconsistent ontologies. Tsinghua Science & Technology **15**(6), 687–691 (2010)
18. Gangemi, A., Sagri, M.T., Tiscornia, D.: A constructive framework for legal ontologies. In: Law and the semantic web, pp. 97–124. Springer (2005)
19. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming **4**(2), 95–138 (2004)
20. Getman, A.P., Karasiuk, V.V.: A crowdsourcing approach to building a legal ontology from text. Artificial intelligence and law **22**(3), 313–335 (2014)
21. Gómez, S.A., Chesñevar, C.I., Simari, G.R.: Reasoning with inconsistent ontologies through argumentation. Applied Artificial Intelligence **24**(1&2), 102–148 (2010)
22. Griffo, C., Almeida, J.P.A., Guizzardi, G.: Towards a legal core ontology based on alexy's theory of fundamental rights. In: Multilingual Workshop on ICAIL (2015)

23. Hoekstra, R., Breuker, J., Di Bello, M., Boer, A., et al.: The lkif core ontology of basic legal concepts. LOAIT **321**, 43–63 (2007)
24. Lu, Y., Yu, Z.: Argumentation theory for reasoning with inconsistent ontologies. In: Borgwardt, S., Meyer, T. (eds.) DL 2020. vol. 2663 (2020)
25. Lu, Y., Yu, Z., Lin, Y., Schafer, B., Ireland, A., Urquhart, L.: Handling inconsistent and uncertain legal reasoning for ai vehicles design. In: Proceedings of Workshop on Methodologies for Translating Legal Norms into Formal Representations (LN2FR 2022) (2022)
26. Lu, Y., Yu, Z., Lin, Y., Schafer, B., Ireland, A., Urquhart, L.: A legal support system based on legal interpretation schemes for ai vehicle designing. In: Proceedings of the 35th International Conference on Legal Knowledge and Information Systems (JURIX 2022). pp. 213–218 (2022)
27. Martinez, M.V., Deagustini, C.A.D., Falappa, M.A., Simari, G.R.: Inconsistency-tolerant reasoning in datalog$^{\pm}$ ontologies via an argumentative semantics. In: IBERAMIA 2014. pp. 15–27 (2014)
28. McLachlan, S., Neil, M., Dube, K., Bogani, R., Fenton, N., Schaffer, B.: Smart automotive technology adherence to the law:(de) constructing road rules for autonomous system development, verification and safety. International Journal of Law and Information Technology **29**(4), 255–295 (2021)
29. Modgil, S., Prakken, H.: A general account of argumentation with preferences. Artificial Intelligence **195**, 361–397 (2013)
30. Poblet, M., Casellas, N., Torralba, S., Casanovas, P.: Modeling expert knowledge in the mediation domain: a middle-out approach to design odr ontologies. In: LOAIT 2009. No. 3è
31. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument & Computation **1**(2), 93–124 (2010)
32. Schlobach, S., Cornet, R., et al.: Non-standard reasoning services for the debugging of description logic terminologies. In: Ijcai. vol. 3, pp. 355–362 (2003)
33. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. Artificial Intelligence **48**(1), 1–26 (1991)
34. Valente, A.: Legal Knowledge Engineering: A Modelling Approach. IOS Press (1995)
35. Valente, A., Breuker, J., Brouwer, B.: Legal modeling and automated reasoning with on-line. International Journal of Human-Computer Studies **51**, 1079–1125 (1999)
36. Van Engers, T., Boer, A., Breuker, J., Valente, A., Winkels, R.: Ontologies in the legal domain. In: Digital Government: E-Government Research, Case Studies, and Implementation. pp. 233–261 (2008)
37. Yu, Z., Lu, Y.: An argumentation-based legal reasoning approach for DL-ontology. arXiv preprint arXiv:2209.03070 (2022), https://arxiv.org/abs/2209.03070
38. Zhang, X., Lin, Z., Wang, K.: Towards a paradoxical description logic for the semantic web. In: FoIKS. pp. 306–325. Springer (2010)

Compliance checking in the energy domain via W3C standards *

Joseph K. Anim[1], Livio Robaldo[1], and Adam Wyner[1]

Swansea University, Swansea, UK
{joseph.anim,livio.robaldo,a.z.wyner}@swansea.ac.uk

Abstract. This paper investigates the use of W3C standards, specifically RDF, OWL, SPARQL, and SHACL, to automate legal compliance checking, in particular, of in-force regulations to extract oil and gas in Ghana. Norms from these regulations exemplify our proposed model: the examples are taken from a sample ontology along with inference rules. The main finding of this paper is that inferences enabled by OWL and SHACL shapes are not expressive enough to represent some existing legal requirements, specifically those imposing constraints on metadata about RDF individuals. To achieve the required expressivity, it is proposed that SHACL-SPARQL rules should be instead used.

Keywords: Compliance checking · W3C standards · symbolic AI

1 Introduction

Due to the ever-growing regulations upon which compliance procedures are conducted, the quantum of documents that companies must submit to prove their compliance with the regulations governing their activities has increased and is increasing in volume[1]. In addition, lawyers mostly check compliance and prepare due diligence documents manually. However, this has several disadvantages: it is highly time-consuming, it is error-prone, and it creates an avenue for corruption as it makes it difficult to understand when errors were either caused by unintentional oversights or they were done on purpose.

LegalTech technologies have been employed to mitigate these problems [3]. Automatizing repetitive operations allows one to save time, enhances accuracy, and makes the whole process easily accountable, which in turn makes corruption less feasible.

Currently, most approaches to compliance checking are based on Machine Learning (ML), see, e.g., [4, 13, 29, 23]. However, ML makes it difficult to handle

* Joseph K. Anim has been supported by the Ghana Scholarship Secretariat (see https://www.scholarshipgh.com). Livio Robaldo has been supported by the Legal Innovation Lab Wales operation within Swansea University's Hillary Rodham Clinton School of Law; the operation has been part-funded by the European Regional Development Fund through the Welsh Government.

[1] https://www.thomsonreuters.com/en-us/posts/investigation-fraud-and-risk/cost-of-compliance-2021

specific and exact values, as it is often required when checking compliance of due diligence documents as well as drawing inferences from the values. The accuracy of ML is intrinsically limited by the Pareto principle, a.k.a., the 80/20 rule; thus it provides results that are correct in *most* cases but not *all* [19]. Furthermore, ML tends to behave like a "black box" unable to explain its decisions; as a consequence, often it is even impossible to distinguish the ∼80% of correct results from the ∼20% of incorrect ones.

To address these limitations, symbolic representations have been proposed and used even though they require more manual efforts [24], [21]. Symbols correspond to human-understandable concepts, thus the chain of logical derivations on symbols could provide intelligible explanations of AI decision-making. To mitigate the fact that symbolic representations require more manual efforts, *standardized formats should be used*, thus allowing the funnelling of efforts from more people, which in turn facilitates reusability and sharing of resources.

This paper presents a methodology for compliance checking based on main W3C standards for the Semantic Web, specifically RDF, OWL, SPARQL, and SHACL. The methodology is exemplified on Ghanaian regulations for extraction of oil and gas, which we will use as case study. It is crucial to research and implement symbolic compliance checkers that are compatible with the mentioned W3C standards because more and more (big) data is becoming available in RDF format. In addition, legal ontologies encoded in RDF/OWL have been increasingly proposed and used within existing LegalTech applications [14, 26]. Legal ontologies specify relevant legal concepts, individuals, constraints, etc. such as duties and rights from legislation, as well as their relationship with the concepts of the domain to which the norms apply, e.g., finance, health, or the energy domain.

As pointed out above, symbolic representations enable human-understandable forms of logical reasoning such as matching the constraints from the ontology with the states of affairs, in order to check whether they comply with the in-force norms. Matching and annotating big data with legislative information will produce even more and richer big data, thus the importance of using the same standardized formats, namely the W3C standards, to achieve interoperability.

The present paper proposes a novel approach to compliance checking using SHACL-SPARQL rules, discussed below. Specifically, in contrast to prior work, e.g., [25], with SHACL shapes and Triple rules, the current work replaces them with SHACL-SPARQL rules which can be used extract metadata, aggregate it, then perform some process on the aggregated information. This is not feasible in prior frameworks. The framework is exercised with respect to an ontology representing regulations in the oil and gas domain.

2 Background - W3C standards for the Semantic Web

As mentioned earlier, the objective of this paper and of our research as a whole is to devise computational methods for compliance checking fully compatible with main W3C standards, in order to foster interoperability with available big data.

W3C has already defined several formats to empower the Semantic Web[2]. This paper will use RDF and OWL to encode the ontology, both the TBox (terminological box), which represents the domain knowledge, and the ABox (assertive box), which represents the states of affairs. Then, we will use SPARQL and SHACL to compute and query new knowledge from the explicitly asserted RDF triples. Specifically, we will model norms as SPARQL and SHACL rules; these rules will be then executed on the states of affairs encoded in RDF to infer which individuals comply with the norms rather than violating them.

2.1 RDF and OWL

RDF (Resource Description Framework) is a language which was primarily instituted to represent information about resources on the World Wide Web. RDF represents metadata about web resources, such as the title, author, and modification date of a web page, etc. However, RDF has evolved such that it can be used to represent any general information about identifiable things on the web. The intent behind RDF was to allow applications to process and exchange information without this information losing its intended meaning. This will ensure that information can be exchanged even between applications that were not developed to use or work with the original information.

RDF is therefore used for creating ontologies, i.e., application-neutral networks of concepts, which are called "RDF resources" (classes, individuals, and properties). RDF includes basic constructs to declare them as well as to relate them to one another.

RDF has been mainly designed to *describe* knowledge. Thus, it has very limited reasoning capabilities. In RDF, it is only possible to infer whether certain RDF resources belong to certain classes via the constructs `rdfs:subClassOf`, `rdfs:domain`, and `rdfs:range`.

OWL (Ontology Web Language) augments RDF by adding more reasoning capabilities. OWL allows specification of many more constraints than RDF, which in turn enable more inferences about the RDF resources. In particular, OWL introduces constructs that allow to infer when certain RDF resources do *not* belong to certain classes, e.g., the construct `owl:disjointWith`. These constructs in particular amplify the reasoning capabilities of the language, which in turn has led to investigations about the trade-off between expressivity and computational complexity of the inferences.

These investigations have identified three main sub-languages of OWL: OWL full, OWL DL, and OWL lite. OWL full and OWL lite feature, respectively, full and very reduced expressivity but, consequently, also full and very reduced computational complexity. OWL DL has intermediate expressivity and complexity between OWL full and OWL lite; DL stands for "Description Logic", the logic that OWL DL refers to[3]. Thus, for applications in which the computational time is relevant, it is advisable to use OWL DL or OWL lite in place of OWL full.

[2] See the list at `https://www.w3.org/2001/sw/wiki/Main_Page`

[3] Description Logic refers to a *family* of logics that are less expressive than First-order Logic; OWL DL more specifically refers to the description logic **SHOIN-D** [12].

Several OWL reasoners have been already proposed in the literature to compute inferred ontologies from the explicitly asserted one, e.g., HermiT[4]. [8] presents a comparison among some of these OWL reasoners. [8] highlights that the reasoners vary significantly with regard to the relevant aims and characteristics, so that each specific case study, especially if in an industrial context, deserves a careful and critical choice of the reasoner to be employed therein.

2.2 SPARQL and SHACL

As part of the Semantic Web activity, the RDF Data Access Working Group released in 2004 the first public working draft of an RDF querying language which was known as SPARQL. Since then, further operators to add, delete, and update the triples in the ontology as well as to deduce new information from them has been added to SPARQL. Nowadays SPARQL is a rich language for both querying and manipulating RDF datasets [17].

SPARQL query are generally embedded and executed within other software or programming languages; examples are the SPARQL plug-in for the Protégé editor and the Jena libraries for Java. Therefore, the order in which SPARQL queries are executed is decided by the user or programmatically in the logic of the software: SPARQL does not provide constructs to relate the queries of one to another, for instance, to establish some execution order on them.

On the other hand, SHACL is a W3C recommendation more recent than SPARQL: it was originally proposed in 2017 for the purpose of validating RDF datasets. SHACL allows to specify special constraints, called "SHACL shapes" on RDF resources. External validators allow to check whether an RDF dataset is valid or not with respect to a set of SHACL shapes. SHACL is more expressive than OWL and it may be therefore used as an alternative to it. In particular, SHACL includes non-monotonic operators such as negation-as-failure. These are not allowed in OWL, which is a monotone language.

Furthermore, SHACL constraints are more flexible and easier to edit than OWL ones because, while the latter are all executed *at once*, in SHACL we may *decouple* complex validation tasks into (simpler) sequential modules. This is possible thanks to the introduction of SHACL rules[5] that enable non-ontological types of operations such as collecting data from RDF resources located in "distant" parts of the ontology or computing partial results needed for the validation [16]. SHACL allows in particular to specify *priorities* on the rules, and so to define sequences or even flow charts of rules, in a rather controlled fashion.

There are two kinds of SHACL rules: SHACL Triple rules, which can add a *single* RDF triple to the inferred ontology, and SHACL-SPARQL rules, which embed SPARQL queries in the form `CONSTRUCT-WHERE`. In SHACL-SPARQL rules, for each subgraph that satisfies the `WHERE` clause, the subgraph in the corresponding `CONSTRUCT` clause is added to the inferred ontology. Therefore, contrary to SHACL Triple rules, SHACL-SPARQL rules may add more than

[4] `http://www.hermit-reasoner.com`
[5] `https://www.w3.org/TR/shacl-af`

one RDF triple to the inferred ontology. Moreover, the expressivity of SHACL-SPARQL rules add to the richness of SPARQL the possibility of establishing *orders* between SPARQL inferences, thus creating the controlled sequences or flow charts of such inferences.

3 Related works

Compliance checking on RDF triples have been already investigated in past literature. Some of the first approaches are [11, 6, 7, 15]. These approaches use RDF/OWL to model the TBox while the states of affairs and separate knowledge bases of legal rules are encoded in special *separated* XML formats such as SWRL[11], LKIF-rules [6], RuleML [7] and LegalRuleML [15].

More recently, but in the same vein, [10] made a preliminary proposal to extend the LegalRuleML meta model [2] and to represent normative rules via SPARQL queries. [10] is, to our knowledge, the first proposal that models normative reasoning by employing W3C standards only. The solution presented in this paper is therefore rather close to [10]'s; however, while [10] only uses SPARQL, our formalization will use SHACL in conjunction with SPARQL.

Another relevant approach is [5], which encodes legal rules within OWL2 decidable profiles in order to keep computational complexity under control. In [5], norms are represented as property restrictions that refer to the subsets of individuals that comply with the norms. Compliance checking is then enforced via OWL2 subsumption. However, the authors themselves acknowledge (see [5], §3.3) that their approach does not really involve legal reasoning, which is defeasible in nature, but it is only limited to GDPR policy validation.

Similarly to [5], [9] and [25] distinguish compliant and non-compliant individuals by introducing, respectively, special OWL subclasses and special SHACL shapes. Contrary to [5], however, [9] and [25] can model defeasible inferences.

For instance, the OWL ontology in [9] include two classes `Supplier` and `Vehicle`. The individuals in `Supplier` are obliged to communicate their contractual conditions to their consumers (R1), while vehicles cannot drive over 90 km/h (R2). To implement R1 and R2, `Supplier` and `Vehicle` respectively include a boolean datatype property `hasCommunicatedConditions` and a float datatype property `hasDrivingSpeed`. Then, two subclasses `SupplierR1compliant` and `VehicleR2compliant` are defined, the former including individuals in `Supplier` for which `hasCommunicatedConditions` is true, the latter including individuals in `Vehicle` for which `hasDrivingSpeed` is lower than 90. Compliance checking is then enforced by simply applying OWL2 subsumption. In other words, OWL "is-a" inferences will populate `SupplierR1compliant` and `VehicleR2compliant` with only the individuals that comply with the two norms.

In this setting, defeasibility may be modelled by defining complement subclasses via the OWL2 tags at disposal, e.g., `owl:disjointWith`. These subclasses will define the subsets of individuals that *violate* the norms; thus, by imposing the set the individuals that comply with a norm as `owl:disjointWith` the one that violate it, the correct inferences are achieved.

The solution in [25] is very close to the one of [9], the crucial difference being that [25] uses SHACL shapes in place of OWL2 subclasses/restrictions to validate the values of the relevant attributes, e.g., `hasCommunicatedConditions` and `hasDrivingSpeed`, in the example above. In [25], SHACL Triple rules are used to compute the values of these attributes. These rules collect partial data from the RDF triples, and so they facilitate the representation of the norms by decoupling it in multiple sequential steps. Once the SHACL Triple rules have computed the values of the attributes, the SHACL shapes are executed in order to validate these values. Individuals with invalid attribute values are labelled as non-compliant. Finally, defeasibility is modeled via negation-as-failure: whenever exceptions hold, corresponding SHACL Triple rules defeat other ones, so that the inferences associated with the latter are blocked.

The next sections will highlight that the approaches in [9] and [25] are both inadequate to represent certain kinds of norm. In particular, in [9] and [25], it is not possible to check compliance on *aggregate data* from the ontology, which are indeed *metadata* about the individuals in the ABox. For example, as described below, we would like to extract metadata (i.e., not specified per se by the ontology) such as the number of Ghanaian technical core employees at a company and the number of all technical core employees at that company, bring them together (aggregate), then use them to *calculate* whether the former is at least 20% of the latter, as required by the relevant regulation. Aggregate data cannot be computed via OWL "is-a" inferences or via SHACL shapes and Triple rules.

On the other hand, to compute aggregate data we need the expressivity offered by SPARQL, in line with the approach presented in [10]. Specifically, this paper will propose a novel revision of the framework in [25] in which SHACL shapes and Triple rules are replaced by SHACL-SPARQL rules in order to extract metadata, aggregate it, then perform some process on the aggregated information. This is not feasible in prior frameworks. The ontology and SHACL-SPARQL rules are used in a compliance checking framework depicted in Figure 1.

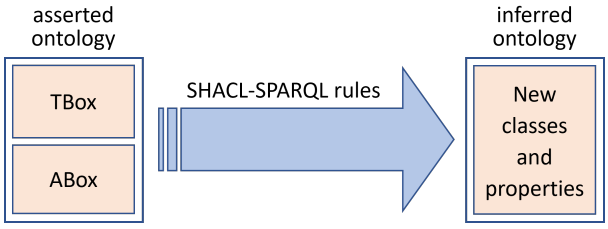


Fig. 1. The compliance checking framework

4 Case study: extracting oil and gas in Ghana

In Ghana, the oil and gas upstream industry has recently seen a lot of foreign investments from international corporations. It is expected these investments will further grow in the near future, due to the conflict in Ukraine and the

consequent need for new alternative sources of oil and gas. The explorations have led to several oil and gas discoveries[6], among which it is worth mentioning the Offshore Cape Three Points (OCTP), which is estimated to hold about 41 billion cubic meters of non-associated gas and 500 million barrels of oil.

These discoveries are in turn expected to greatly contribute to the Ghanaian gross domestic product. It has been estimated that the oil and gas industry will contribute approximately 15.94 billion GHS (around 2.76 billion U.S. dollars) to Ghana's gross domestic product in 2024 [27].

As a result, the upstream oil and gas industry has become one of the heavily regulated sectors in Ghana. Companies operating in this domain are required to submit several due diligence documents to auditing agencies such as the Ghana Petroleum Commission in order to check for regulatory compliance with respect to the in-force regulations[7]. In this work, we focus on the Local Content and Local Participation Regulations L.I 2204, henceforth named as "L.I 2204" only.

The L.I 2204 aim at ensuring the participation of Ghanaians and the use of indigenous materials in the upstream oil and gas industry. Simply put, the L.I 2204 are intended to prevent foreign companies from bringing their own employees and materials from abroad. Rather, they are allowed to extract the country's oil and gas only on condition they create employment and other economical benefits for the local population.

In particular, the regulation 7(2)(B) of the L.I 2204 requires companies in the upstream oil and gas industry to provide annually a "Local Content Plan", which includes several sub-plans (Employment Plan, Training plan, Insurance services plan, etc.) and which the company specifies information about the impact of the company's business into the Ghanaian local economy.

The overall aim of our work is to design and implement a LegalTech application able to assist the compilation and the assessment of the Local Content Plan. The present paper represents the first step of this research journey: it aims to present a first prototype of an ontology that can be used to collect and store data about companies in the upstream oil and gas industry then used to automatically check their compliance with the L.I 2204. That is, companies are expected to use a Web interface to the Local Content Plan and determine what their obligations with respect to the L.I 2204. Further data could be integrated in the ontology from other Ghanaian institutions and sources, e.g., the chamber of commerce, and double-checked against the information entered by the company. These double-checks, not implemented in our current work, are of course intended to detect (possibly unintentional) oversights and errors, as well as to speed up and assist the compilation of the Local Content Plan by self-inserting the data already known. In our prototype, companies are expected to enter data about the bank supporting their financial operations, the law firm that is assisting their business, as well as their employees.

Some of the legal requirements that the Local Content Plan is intended to assess are the following:

[6] See `https://www.gnpcghana.com/operations.html`

[7] Listed at `https://www.petrocom.gov.gh/laws-regulations`

(1) a. Is the company banking with a Ghanaian bank?
b. Is the company hiring the legal services of a Ghanaian law firm?
c. Is the company employing at least 30% of Ghanaian management staff?
d. Is the company employing at least 20% of Ghanaian technical core staff?
e. Is the company employing 100% of Ghanaian middle or junior level staff?
f. Etc.

We created a small ontology including some of the relevant classes and properties (TBox) from our domain. We populated the ontology with some sample individuals and relations between them (ABox). Then, we modeled some sample legal requirements, among which those in (1.a-e), as SHACL-SPARQL rules.

These rules create additional classes and properties to distinguish compliant and non-compliant individuals, and populate the classes with these individuals. By executing the rules on the (asserted) ontology, a new (inferred) ontology is obtained. The latter will therefore represent which companies comply or not with the modeled legal requirements as well as the *explanations* why they do or do not comply with these requirements.

The hypothetical LegalTech application for the Local Content Plan will query the inferred ontology via simple SPARQL queries, in order to generate a report of the compliance assessment. Note that the inferred triples are not saved and stored together with the original asserted ontology. The additional classes and properties have the sole purpose of classifying the companies as compliant or non-compliant. Once these have been identified and communicated to the LegalTech application, the inferred ontology is simply discharged.

The next two subsections illustrate part of the asserted ontology and some SHACL-SPARQL queries[8].

4.1 The (asserted) ontology

We modeled the domain of the Local Content Plan as an ontology in OWL. The ontology includes classes referring to the sets of relevant entities. Some of these classes are shown in Figure 2.

Figure 2 shows the classes of legal entities, activities, areas, and structures involved in the modeled legal requirements as well. For example, `SectorCompany` denotes the set of all companies in the upstream oil and gas industry operating in Ghana. Individuals from one class can be related to individuals from another class by object properties. For instance, each individual in `Employee` is related with an individual in `Nationality` via an object property "`is`"; `Nationality` is a value partition including the individuals `ghanaian`, `italian`, `american`, etc. On the other hand, individuals in `SectorCompany` are associated with information specifying the areas they operate, which activities they carry out in these areas, which structures they used within these activities, the type of gas (`methane`, `propane`, `butane`, etc.) they work with, etc.

[8] The full ontology and list of queries is available on `https://github.com/liviorobaldo/jurisin2023ca`, together with Java software to execute the latter on the former thus obtaining the inferred ontology.

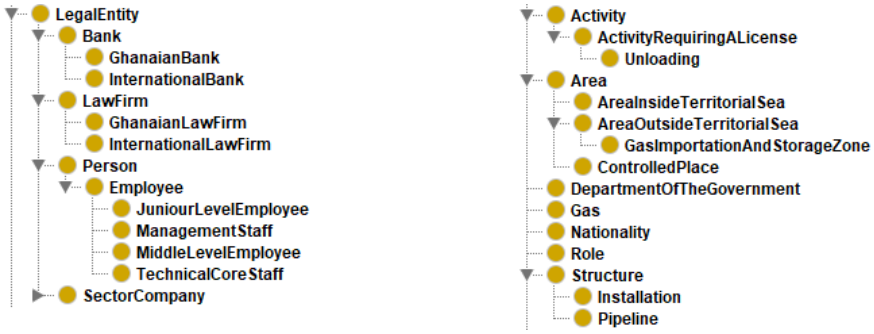


Fig. 2. Some of the classes in the ontology for the Local Content Plan

The ontology have been then populated with sample individuals in order to test the SHACL-SPARQL rules. Figure 3 shows some of these individuals and object properties (e.g., `employs`, `is`, `bank-with`). Two sample companies are considered: `companyc` and `companye`. The former banks with a Ghanaian bank while the latter banks with an international bank. Furthermore, `companyc` employs four technical core employees, having all Ghanaian nationality, while `companye` employs two technical core employees, having respectively Italian and American nationality.

It is then evident that `companyc` complies with legal requirements (1.a) and (1.d) while `companye` violates both of them. The SHACL-SPARQL rules described in the next section allows to infer these compliance checking results.

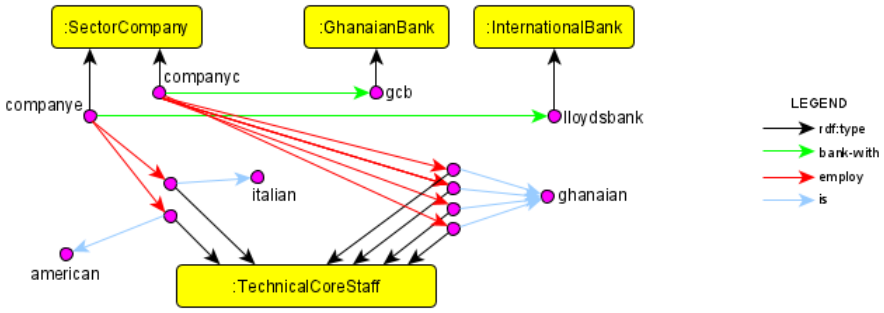


Fig. 3. Sample individuals for the case study

4.2 The SHACL/SPARQL rules

The previously referenced computational artefact includes around twenty-nine SHACL-SPARQL queries to implement some selected legal requirements from the L.I 2204. Space constraints limit showing and describing all of them in detail; we will focus only on the ones that implement (1.a) and (1.d).

The two SHACL-SPARQL rules that implement (1.a) are shown together in (2). The rules embed a SPARQL query in the form `CONSTRUCT-WHERE` within the SHACL property `sh:construct`. As specified in the `WHERE` clause, the first rule collects all companies that bank with a Ghanaian bank (`?x :bank-with ?y. ?y rdf:type :GhanaianBank.`). The `CONSTRUCT` clause:

- creates then a *new* class `BLCCompliantSectorCompany` in the inferred ontology (`:BLCCompliantSectorCompany rdf:type rdfs:Class.`)
- asserts `BLCCompliantSectorCompany` as subclass of the class `SectorCompany` (`:BLCCompliantSectorCompany rdfs:subClassOf :SectorCompany.`)
- asserts the individuals `?x` that satisfy the `WHERE` clause as instances of this new class (`?x rdf:type :BLCCompliantSectorCompany.`)

The second rule in (2) is very similar to the first one. The rule collects all individuals that bank with an international bank and asserts them as instances of a newly created class `BLCNonCompliantSectorCompany`.

Thus, the two rules together distinguish individuals that comply with (1.a) from those that do not: the LegalTech application will query the inferred ontology by listing all individuals belonging to either `BLCCompliantSectorCompany` or `BLCNonCompliantSectorCompany` via simple SPARQL queries.

(2)
```
sh:rule [rdf:type sh:SPARQLRule;
  sh:prefixes[sh:declare
    [sh:prefix"rdf";sh:namespace"..."],
    [sh:prefix"rdfs";sh:namespace"..."], ... ];
  sh:construct """
    CONSTRUCT{ :BLCCompliantSectorCompany rdf:type rdfs:Class.
           :BLCCompliantSectorCompany rdfs:subClassOf :SectorCompany.
           ?x rdf:type :BLCCompliantSectorCompany. }
    WHERE{ ?x :bank-with ?y.
           ?y rdf:type :GhanaianBank. }""";];

sh:rule [rdf:type sh:SPARQLRule;
  sh:prefixes[sh:declare
    [sh:prefix"rdf";sh:namespace"..."],
    [sh:prefix"rdfs";sh:namespace"..."], ... ];
  sh:construct """
    CONSTRUCT{ :BLCNonCompliantSectorCompany rdf:type rdfs:Class.
        :BLCNonCompliantSectorCompany rdfs:subClassOf :SectorCompany.
        ?x rdf:type :BLCNonCompliantSectorCompany. }
    WHERE{ ?x :bank-with ?y.
           ?y rdf:type :InternationalBank. }""";]
```

Although the rules in (2) employ a different technology than [9]'s and [25]'s, the expressivity and the "modus operandi" of the three approaches is exactly the same. In other words, [9] and [25] are also designed to populate two classes such as `BLCCompliantSectorCompany` and `BLCNonCompliantSectorCompany` with all individuals that respectively comply with or not with (1.a).

By contrast, it is not possible to implement the legal requirement (1.d) via OWL classes/restrictions, as in [9], or via SHACL shapes, as in [25]. (1.d) requires to *count* both the number of Ghanaian technical core employees and the number of all technical core employees, and then to *calculate* whether the former is at least 20% of the latter. OWL "is-a" inferences and SHACL shapes are not expressive enough to query *metadata* of RDF individuals. On the contrary, SPARQL offers the desired expressivity thanks to its *aggregate functions*[9] and its *arithmetic functions*[10].

However, SPARQL alone is not enough to implement (1.d) because the two operations of counting the sets of technical core employees and calculating the proportion among these sets cannot be done via a single rule. SHACL provides the missing ingredient by allowing to *decouple* the implementation of the legal requirement into two *sequential* rules. By using SHACL-SPARQL rules as proposed here, the problem can be addressed.

The two SHACL-SPARQL rules that count the number of Ghanaian technical core employees and the total number of such employees are shown in (3).

(3)
```
sh:rule [rdf:type sh:SPARQLRule; sh:order 0;
  sh:prefixes[sh:declare
    [sh:prefix"rdf";sh:namespace"..."],
    [sh:prefix"rdfs";sh:namespace"..."], ... ];
  sh:construct """
    CONSTRUCT {?x :gh_tec_emp ?gh_tec_emp.}
    WHERE{ SELECT ?x (count(?y) as ?gh_tec_emp)
           WHERE{ ?x rdf:type :SectorCompany.
                  ?x :employ ?y.
                  ?y rdf:type :TechnicalCoreStaff.
                  ?y :is :ghanaian.} GROUP BY ?x}"""]

sh:rule [rdf:type sh:SPARQLRule; sh:order 0;
  sh:prefixes[sh:declare
    [sh:prefix"rdf";sh:namespace"..."],
    [sh:prefix"rdfs";sh:namespace"..."], ... ];
  sh:construct """
    CONSTRUCT {?x :tec_emp ?tec_emp.}
    WHERE{ SELECT ?x (count(?y) as ?tec_emp)
           WHERE{ ?x rdf:type :SectorCompany.
                  ?x :employ ?y.
                  ?y rdf:type :TechnicalCoreStaff.} GROUP BY ?x}"""]
```

In (3), `sh:order` is the SHACL operator to order the rules. These are executed from the lowest value of `sh:order` to the highest one. The two rules associate every sector company `?x` with, respectively, their numbers of Ghanaians technical core employees and their number of overall technical core employees via two newly created datatype properties `gh_tec_emp` and `tec_emp`.

[9] https://en.wikibooks.org/wiki/SPARQL/Aggregate_functions
[10] https://en.wikibooks.org/wiki/SPARQL/Expressions_and_Functions

A separate rule, shown in (4) and executed *after* the ones in (3), because its `sh:order` is equal to 1, calculates the proportion between the values of the datatype properties `gh_tec_emp` and `tec_emp`, asserted via the previous rules. (4) asserts all individuals for which the proportion is lower than 20% as instances of a newly created class `Nc_Gh_Tec_Emp`.

(4)
```
sh:rule [rdf:type sh:SPARQLRule; sh:order 1;
  sh:prefixes[sh:declare
    [sh:prefix"rdf";sh:namespace"..."],
    [sh:prefix"rdfs";sh:namespace"..."], ... ];
  sh:construct """
  CONSTRUCT{ :Nc_Gh_Tec_Emp rdf:type rdfs:Class.
             ?x rdf:type :Nc_Gh_Tec_Emp. }
  WHERE{ ?x rdf:type :SectorCompany.
         ?x :gh_tec_emp ?gh_tec_emp.
         ?x :tec_emp ?tec_emp.
         FILTER(?gh_tec_emp<(?tec_emp*0.2)). }"""]
```

Finally, the LegalTech application can again retrieve the list of individuals belonging to the class `Nc_Gh_Tec_Emp`, i.e., the list of sector companies that do not comply with (1.d), via a simple SPARQL query.

4.3 comparison to existing techniques for compliance checking

There are a few legal tech approaches or techniques which have been adopted to deal with compliance checking. However, for the purposes of this paper, we will briefly talk about the machine learning approach particularly, the Machine Learning techniques which were used in the revelation of VAT Compliance Violations in Accounting Data and compare it to our approach. As already posited in this paper, one way of going about compliance checking is employing machine learning techniques. However, since our goal for this paper is to be able to exert, respect and follow logical patterns in compliance checking in order to achieve explanability, it is our hamble opinion that machine learning is inadequate for this purpose especially when machine learning is based on statistical rather than logical reasoning. Having established this, we will proceed to describe the machine learning technique mentioned above. the authors of the article, "Utilizing Machine Learning Techniques to Reveal VAT Compliance Violations in Accounting Data" adopted some machine learning techniques such as training a multi-class classifiers to be able to accurately determine tax code for an unseen transaction, among many others. The aim of this project was to present a compliance system for ERP systems which will, with the aid of machine learning methods, identify the nonconformity to compliance regulations in course of the process of determining the VAT tax code. To materialize their aim, they undertook the training of machine learning classification models which were ultimately capable of detecting anomalies particularly, identifying obvious transactions which were most probable to be assigned to a false tax code. Lahann-etal:19 Comparing this particular approach to our approach, just as stated supra, this approach involves

training of machine learning models to achieve a specific level of accuracy. I.E., statistical reasoning is used to attain this level of accuracy through training. Thus, unlike our approach to compliance checking, logical reasoning is not involved and so no matter how accurate the compliance system is, its results I.E., the violation or other wise of the regulations cannot be explained. Also, based on the black box theory of machine learning, the approach adopted in this case cannot assist in tracing the means through which a particular decision is made. However, as demonstrated in our approach, because of the inherent exertion of logic in the representation of the knowledge base, every single decision reached by a legal tech system premised on our knowledge base can be explained and the means through which the decision was reached can be traced. Our approach is more in synch with legal practice because in the law every decision must be explained or is capable of being explained. That is why judges or even regulatory bodies provide reasons for reaching a particular decision.

5 Conclusions

This paper contributes the means to automatise compliance checking to Semantic Web technologies. The main motivation behind researching solutions grounded on W3C standards is the hypothesis that, in the future, these standards will likely serve as the basis of symbolic explainable Artificial Intelligence, particularly for legal applications.

Some recent approaches along these lines, e.g., [5] and [9], propose solutions for compliance checking based on OWL2 inferences; the main motivation behind this technological choice is to keep the framework *decidable*.

Although controlling computational complexity is of course crucial, it should be privileged over the expressivity of the inferences only when there is really no other way to make the application working in reasonable time.

This paper provided evidence that OWL2 inferences do not seem to be enough expressive for representing several compliance checks required by existing regulations, specifically those checking and aggregating *metadata* of RDF individuals.

Subsection 4.2 above exemplified this kind of checks out of a real-world legal requirement that we found in the Local Content and Local Participation Regulations for extracting oil and gas in Ghana. Companies in the oil and gas upstream industry are required to employ at least 20% of Ghanaian technical core staff. In order to represent this requirement, we had to use SHACL-SPARQL rules: SPARQL provides aggregate and arithmetic operators, while SHACL allows to build sequences or flow charts of operations by specifying priorities on the rules.

Although we have not conducted (yet) any empirical investigation about the frequency of this kind of norms in existing regulations, we indeed believe they are rather frequent. Regulations often impose legal constraints on, for instance, *sums* of money or *minimum*/*maximal* numerical values, or they require to *count* number of days/requests/attempts/etc., or to *check dates*, etc. All these constraints requires to process metadata of RDF individual. Therefore, their implementa-

tion requires the same expressivity offered by SHACL-SPARQL rules but not by OWL or SHACL shapes and Triple rules.

On the other hand, the computational complexity of SPARQL and SHACL does not seem to be significantly problematic (cf. [17, 1]), nor, more generally, the one of compliance checkers based on sets of explicit if-then rules (cf. [28]). In fact, SHACL-SPARQL rules may be easily converted into other rule-based logical languages such as Answer Set Programming [20] [22], for which automated reasoners with very good computational performance are available.

In future work, we will further enrich the ontology and evaluate it. An additional important direction is to incorporate time management in the ontology, for which we are planning to import existing ontologies such as OWL-Time[11] and the Time-indexed Value in Context ontology [18].

References

1. Ahmetaj, S., David, R., Ortiz, M., Polleres, A., Shehu, B., Šimkus, M.: Reasoning about Explanations for Non-validation in SHACL. In: Proc. of 18th International Conference on Principles of Knowledge Representation and Reasoning (2021)
2. Athan, T., Boley, H., Governatori, G., Palmirani, M., Paschke, A., Wyner, A.: LegalRuleML: From Metamodel to Use Cases. In: Morgenstern, L., Stefaneas, P., Lévy, F., Wyner, A., Paschke, A. (eds.) Theory, Practice, and Applications of Rules on the Web. Springer Berlin Heidelberg (2013)
3. Boella, G., Caro, L.D., Humphreys, L., Robaldo, L., Rossi, P., van der Torre, L.: Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law. Artificial Intelligence and Law **24**(3) (2016)
4. Boella, G., Caro, L.D., Rispoli, D., Robaldo, L.: A system for classifying multi-label text into eurovoc. In: Proc. of the International Conference on Artificial Intelligence and Law (ICAIL). ACM (2013)
5. Bonatti, P.A., Ioffredo, L., Petrova, I.M., Sauro, L., Siahaan, I.S.R.: Real-time reasoning in OWL2 for GDPR compliance. Artificial Intelligence **289** (2020)
6. Ceci, M.: Representing judicial argumentation in the semantic web. In: Casanovas, P., Pagallo, U., Palmirani, M., Sartor, G. (eds.) AI Approaches to the Complexity of Legal Systems. Lecture Notes in Computer Science, vol. 8929. Springer (2013)
7. De Vos, M., Kirrane, S., Padget, J.A., Satoh, K.: ODRL policy modelling and compliance checking. In: Fodor, P., Montali, M., Calvanese, D., Roman, D. (eds.) Rules and Reasoning - Third International Joint Conference, RuleML+RR (2019)
8. Dentler, K., Cornet, R., ten Teije, A., de Keizer, N.: Comparison of reasoners for large ontologies in the OWL 2 EL profile. Semantic Web **2**(2), 71–87 (2011)
9. Francesconi, E., Governatori, G.: Patterns for legal compliance checking in a decidable framework of linked open data. Artificial Intelligence and Law (2022)
10. Gandon, F., Governatori, G., Villata, S.: Normative requirements as linked data. In: Wyner, A.Z., Casini, G. (eds.) Legal Knowledge and Information Systems. vol. 302. IOS Press (2017)
11. Gordon, T.F.: Constructing legal arguments with rules in the legal knowledge interchange format (LKIF). In: Casanovas, P., Sartor, G., Casellas, N., Rubino, R. (eds.) Computable Models of the Law. Springer (2008)

[11] `https://www.w3.org/TR/owl-time`

12. Horrocks, I.: OWL: A description logic based ontology language. In: van Beek, P. (ed.) Proc. of 11th Int. Conference on Principles and Practice of Constraint Programming. Lecture Notes in Computer Science, vol. 3709. Springer (2005)
13. Nanda, R., Caro, L.D., Boella, G., Konstantinov, H., Tyankov, T., Traykov, D., Hristov, H., Costamagna, F., Humphreys, L., Robaldo, L., Romano, M.: A unifying similarity measure for automated identification of national implementations of european union directives. In: Proc. of the International Conference on Artificial Intelligence and Law (ICAIL). ACM (2017)
14. Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L.: Pronto: Privacy ontology for legal compliance. In: 18^{th} EU conference on Digital Government (2018)
15. Palmirani, M., Governatori, G.: Modelling legal knowledge for GDPR compliance checking. In: 31st Conference on Legal Knowledge and Information Systems (2018)
16. Pareti, P., Konstantinidis, G., Norman, T.J., Sensoy, M.: SHACL constraints with inference rules. In: 18th International Semantic Web Conference. Springer (2019)
17. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of sparql. ACM Transactions on Database Systems **34**(3) (2009)
18. Peroni, S.: The Semantic Publishing and Referencing Ontologies. Springer International Publishing (2014)
19. Reed, W.J.: The pareto, zipf and other power laws. Economics Letters **74**(1) (2001)
20. Robaldo, L., Batsakis, S., Calegari, R., Calimeri, F., Fujita, M., Governatori, G., Morelli, M., Pacenza, F., Pisano, G., Satoh, K., Tachmazidis, I., Zangari, J.: Compliance checking on first-order knowledge with conflicting and compensatory norms - a comparison among currently available technologies. Artificial Intelligence and Law (to appear) (2023)
21. Robaldo, L., Caselli, T., Russo, I., Grella, M.: From italian text to timeml document via dependency parsing. In: Gelbukh, A. (ed.) Proc. of 12th Int. conference "Computational Linguistics and Intelligent Text Processing". Lecture Notes in Computer Science, vol. 6609. Springer (2011)
22. Robaldo, L., Pacenza, F., Zangari, J., Calegari, R., Calimeri, F., Siragusa, G.: Efficient compliance checking of rdf data. Journal of Logic and Computation (to appear) (2023)
23. Robaldo, L., Villata, S., Wyner, A., Grabmair, M.: Introduction for artificial intelligence and law: special issue "natural language processing for legal texts". Artificial Intelligence and Law **27**(2) (2019)
24. Robaldo, L.: Distributivity, collectivity, and cumulativity in terms of (in)dependence and maximality. Journal of Logic, Language and Information **20**(2) (2011)
25. Robaldo, L.: Towards compliance checking in reified I/O logic via SHACL. In: Maranhão, J., Wyner, A.Z. (eds.) Proc. of 18th International Conference for Artificial Intelligence and Law (ICAIL 2021). ACM (2021)
26. Robaldo, L., Bartolini, C., Palmirani, M., Rossi, A., Martoni, M., Lenzini, G.: Formalizing GDPR provisions in reified I/O logic: The DAPRECO knowledge base. Journal of Logic, Language, and Information **29**(4) (2020)
27. Sasu, D.: Oil and gas sector contribution to GDP in Ghana 2014-2024. In: Statista, available at `https://www.statista.com/statistics/1235708/gdp-of-the-oil-and-gas-industry-in-ghana` (2021)
28. Sun, X., Robaldo, L.: On the complexity of input/output logic. The Journal of Applied Logic **25**, 69–88 (2017)
29. Zhang, R., El-Gohary, N.: A machine learning approach for compliance checking-specific semantic role labeling of building code sentences. In: Proc. of the 35th International Conference of IT in Design, Construction, and Management (2018)

Constructing and Explaining Case Models: A Case-based Argumentation Perspective

Wachara Fungwacharakorn[1][0000−0001−9294−3118], Ken Satoh[1][0000−0002−9309−4602], and Bart Verheij[2][0000−0001−8927−8751]

[1] National Institute of Informatics and SOKENDAI, Tokyo, Japan
{wacharaf,ksatoh}@nii.ac.jp
[2] Artificial Intelligence, University of Groningen, Groningen, The Netherlands
bart.verheij@rug.nl

Abstract. In this paper, we investigate constructing and explaining case models, which have been proposed as formal models for presumptive reasoning and evaluating arguments from cases. Recent research shows applications of case models and relationships between case models and other computational reasoning models. However, formal methods for constructing and explaining case models have not been investigated yet. Therefore, in this paper, we present methods for constructing and explaining case models based on the formalism of abstract argumentation for case-based reasoning (AA-CBR). The methods are illustrated in this paper with a legal example of paying penalties for a delivery company. We found that constructed case models can provide model-theoretic semantics equivalent to AA-CBR, and that explanations in case models can be made by dispute trees as in AA-CBR.

Keywords: case-based reasoning · argumentation frameworks · case models

1 Introduction

Artificial Intelligence and Law researchers are interested in explanations of reasons using cases. In order to explain reasons, early case-based legal reasoning systems, such as HYPO, use analogical reasoning [4]. Later, argumentation has been shown to be useful for explanation [3], and case-based argumentation has been shown to be useful not only for explaining from precedent cases [7], but also explaining the development of case law [9] as well as explaining legal theories based on hypothetical cases in statute law [13].

Also, Artificial Intelligence and Law researchers are interested in evaluations of arguments using cases. Case models [16] have been recently developed in order to formally evaluate arguments. Each case model consists of a set of consistent, mutually incompatible, and different logic formulas representing cases, and a total and transitive preference ordering over the cases. Case models evaluate arguments as incoherent, coherent, presumptively valid, and conclusive. Several applications of case models have been investigated, including evidential reasoning

[10, 17] and ethical system design [15]. Formalizing case models for case-based reasoning has also been investigated [15, 16]. However, the questions of how to formally construct case models from a case-base and how to explain argument moves in case models have not been studied.

In order to address these questions, we investigate case models from a case-based argumentation perspective. As a representative of case-based argumentation, we consider an abstract argumentation for case-based reasoning (AA-CBR) [6], which inspires explanations in precedential constraint [12, 18]. A case-base in AA-CBR is a finite set of case-pairs – including a default case-pair, which is a pair of a default situation represented as the empty set and a predefined default outcome. Given a new situation, AA-CBR infers an outcome by forming a corresponding argumentation framework [8] with respect to the case-base, and determining whether or not the default case-pair is included in the grounded extension of the corresponding argumentation framework. AA-CBR explains the inference using dispute trees with respect to the default case-pair. By exploring the relation between AA-CBR and case models, we present a method of constructing case models from an AA-CBR case-base. Furthermore, we extend several concepts in case models to explain argument moves in case models using dispute trees as in AA-CBR. We show that dispute trees used in case models are homomorphic to dispute trees used in AA-CBR.

This paper is structured as follows. Section 2 describes an abstract argumentation for case-based reasoning (AA-CBR), which is a representative of case-based argumentation formalism used in this paper. Section 3 describes case models. Section 4 presents the first contribution of formalizing a method for constructing case models from AA-CBR case-bases. Then, Section 5 presents the second contribution of developing dispute trees for explanations in case models. Section 6 discusses connections with related research, and provides suggestions for future work. Finally, Section 7 provides the conclusion of this paper.

2 Abstract Argumentation for Case-based Reasoning

In this section, we account for an abstract argumentation for case-based reasoning (AA-CBR) [6], which is used as a representative of case-based argumentation in this paper. AA-CBR aims for formalizing reasoning from consistent case-bases with default outcomes. AA-CBR uses Dung's abstract argumentation frameworks [8], which we recap here as follows (cf. [6]).

Definition 1 (AA-framework). *An* AA-framework *is a pair* $\langle AR, attacks\rangle$*, where* AR *is a set whose elements are called* arguments*, and* $attacks \in AR \times AR$*. For arguments* $x, y \in AR$*, if* $(x, y) \in attacks$*, then we say* x attacks y*. For a set* $E \subseteq AR$ *and arguments* $x, y \in AR$*, we say* E attacks x *if some argument* $z \in E$ *attacks* x*; and we say* E defends y *if, for all arguments* $x \in AR$ *that attack* y*,* E *attacks* x*. The* grounded extension *of* $\langle AR, attacks\rangle$ *refers to a set* $G \subseteq AR$ *that can be constructed inductively as* $G = \bigcup_{i \geqslant 0} G_i$*, where* G_0 *is the set of unattacked arguments, and* $\forall i \geqslant 0$*,* G_{i+1} *is the set of arguments that* G_i *defends.*

Dispute trees for arguments in an AA-framework are defined as follows [6].

Definition 2 (Dispute Tree). *Let $\langle AR, attacks\rangle$ be an AA-framework. A* dispute tree *for an argument $x_0 \in AR$, is a (possibly infinite) tree $\mathcal{T}$ such that:*

1. *every node of $\mathcal{T}$ is of the form $[L : x]$, with $L \in \{P, O\}$ and $x \in AR$ where L indicates the status of proponent (P) or opponent (O);*
2. *the root of $\mathcal{T}$ is $[P : x_0]$;*
3. *for every proponent node $[P : y]$ in $\mathcal{T}$ and for every $x \in AR$ such that x attacks y, there exists $[O : x]$ as a child of $[P : y]$;*
4. *for every opponent node $[O : y]$ in $\mathcal{T}$, there exists at most one child of $[P : x]$ such that x attacks y;*
5. *there are no other nodes in $\mathcal{T}$ except those given by 1-4.*

A dispute tree $\mathcal{T}$ is an admissible *dispute tree if and only if (i) every opponent node $[O : x]$ in $\mathcal{T}$ has a child, and (ii) no $[P : x]$ and $[O : y]$ in $\mathcal{T}$ such that $x = y$. A dispute tree $\mathcal{T}$ is a* maximal *dispute tree if and only if for all opponent nodes $[O : x]$ which are leaves in $\mathcal{T}$ there is no argument $y \in AR$ such that y attacks x.*

Admissible dispute trees are maximal dispute trees but not vice versa [6] because admissible dispute trees are those maximal dispute trees without opponent leaves while maximal dispute trees with opponent leaves also exist. In other words, admissible dispute trees demonstrate argumentations where the proponent can attack all of the opponent's arguments but maximal dispute trees demonstrate argumentations where the proponent's burden is *complete*, i.e. either the proponent cannot attack some opponent's arguments or the proponent already attacks all of the opponent's arguments.

Recently, researchers [5] have generalized AA-CBR for more general representations of situations and preferences. However, in this paper, we mostly follow definitions from the original work [6]. Let $\mathcal{F}$ be a set of propositions called a *fact-domain*, whose elements are called *fact-propositions* [3]. We call a finite subset of $\mathcal{F}$ a *fact-situation*. Let $o \in \{+, -\}$ be an *outcome*. We denote the opposite of $o \in \{+, -\}$ by $\bar{o}$, namely $\bar{o} = +$ if $o = -$; and $\bar{o} = -$ if $o = +$. A *case-pair* is a pair $(X, o) \in 2^{\mathcal{F}} \times \{+, -\}$. A *case-base* in AA-CBR is then defined as follows [6].

Definition 3 (Case-base in AA-CBR). *A* case-base *is a finite set $CB \subseteq 2^{\mathcal{F}} \times \{+, -\}$ of cases-pairs such that:*

- *(consistent) for $(X, o_x), (Y, o_y) \in CB$, if $X = Y$, then $o_x = o_y$*
- *(containing a default case-pair) $(\varnothing, d) \in CB$, $(\varnothing, d)$ is then called a* default *case-pair and d is called a* default *outcome*

Example 1. To illustrate case-based argumentation, we adapt an example of penalties from a delivery company [2] with the following rules.

[3] In the original work, those elements are called *factors* but we use the new terms in order to distinguish them from factors in CATO [1]

1. If there is no special situation, the delivery company does not have to pay a penalty.
2. If the delivery was delayed, the delivery company has to pay a penalty.
3. If the items were damaged, the delivery company has to pay a penalty.
4. If the items were damaged but they are fungible and the items were replaced, then the delivery company does not have to pay a penalty.

We represent propositions as follows.

- `delayed`: the delivery was delayed.
- `damaged`: the items were damaged.
- `fungible`: the items are fungible
- `replaced`: the items were replaced.
- `penalty`; the delivery company has to pay a penalty

Considering a conclusion of whether the delivery company has to pay a penalty (+ means the company has to pay a penalty; − otherwise), the working example can be represented as a case-base consisting of the following case-pairs.

1. $co_0 = (\varnothing, -)$
2. $co_1 = (\{\texttt{delayed}\}, +)$
3. $co_2 = (\{\texttt{damaged}\}, +)$
4. $co_3 = (\{\texttt{damaged}, \texttt{fungible}, \texttt{replaced}\}, -)$

To infer an outcome for a new fact-situation $N \subseteq \mathcal{F}$, AA-CBR forms an AA-framework and considers whether or not a default case-pair $(\varnothing, d)$ is in the grounded extension of the formed AA-framework.

Definition 4 (AA-framework used in AA-CBR). *An AA-framework corresponding to a case-base CB with a default case-pair $(\varnothing, d)$ and a new fact-situation $N \subseteq \mathcal{F}$ is $\langle AR, attacks\rangle$ satisfying the following conditions:*

- $AR = \{(X, o) \in CB | X \subseteq N\}$ [4]
- (X, o_x) *attacks* (Y, o_y) *for all case-pairs* (X, o_x), $(Y, o_y) \in AR$ *such that*
  - *(different outcomes)* $o_x \neq o_y$ *, and*
  - *(specificity)* $Y \subsetneq X$*, and*
  - *(concision)* $\nexists (Z, o_x) \in AR$ *with* $Y \subsetneq Z \subsetneq X$

The AA-outcome *of N is d if $(\varnothing, d)$ is in the grounded extension of $\langle AR, attacks\rangle$, otherwise the* AA-outcome *of N is $\bar{d}$.*

Another concept discussed in AA-CBR is a nearest case-pair. which is defined as follows [7].

Definition 5 (Nearest case-pair). *Let $N \subseteq \mathcal{F}$ be a fact-situation, and CB be a case-base. $(X, o_x) \in CB$ is (possibly not unique) nearest to N if and only if $X \subseteq N$, and $\nexists (Y, o_y) \in CB$ with $Y \subseteq N$ and $X \subsetneq Y$. In other words, X is $\subseteq$-maximal in the case-base.*

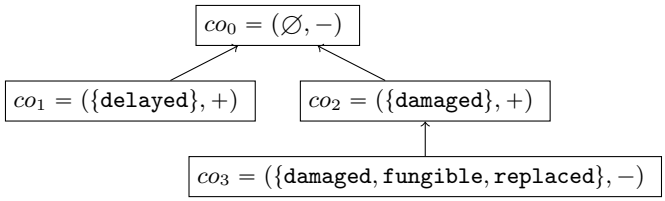


Fig. 1. The AA-framework corresponding to the case-base and N_2 in Example 2

One property of a nearest case-pair is that: if $(X, o_x) \in CB$ is a unique nearest case-pair to a fact-situation $N \subseteq \mathcal{F}$, then the AA-outcome of N is o_x [7].

Example 2. From Example 1, suppose we would like to infer an outcome for a situation where items were damaged, the damaged items are fungible, but the delivery company did not replace the items. The situation can be represented as $N_1 = \{\texttt{damaged}, \texttt{fungible}\}$ and the arguments in the AA-framework corresponding to the case-base and N_1 are co_0 and co_2. We have that co_2 is a unique nearest case-pair to N_1 hence the AA-outcome of N_1 is $+$.

Now, suppose we would like to infer an outcome for a situation where items were damaged, the damaged items are fungible, the items were replaced, but the delivery was delayed. The situation can be represented as $N_2 = \{\texttt{damaged}, \texttt{fungible}, \texttt{replaced}, \texttt{delayed}\}$ and the arguments in the AA-framework corresponding to the case-base and N_2 are all case-pairs in the case-base, as depicted in Figure 1. For this situation, there is no unique nearest case-pair to N_2. To resolve this, we need to consider the grounded extension of the AA-framework. We can see that the default case-pair $co_0 = (\varnothing, -)$ is not in the grounded extension of the AA-framework. Thus, the AA-outcome of N_1 is $+$, i.e. the delivery company has to pay a penalty.

For explanations, AA-CBR uses dispute trees as follows.

Definition 6 (AA-CBR explanation). *Let N be a fact-situation and d be a default outcome. An* explanation *for why the AA-outcome of N is d is any admissible dispute tree for $(\varnothing, d)$. An* explanation *for why the AA-outcome of N is $\bar{d}$ is any maximal dispute tree for $(\varnothing, d)$.*

We refer to a case-pair that can occur in any maximal dispute tree for $(\varnothing, d)$ as a *critical* case-pair (cf. [13]). As a result, $(\varnothing, d)$ is a critical case-pair and any case-pair that attacks a critical case-pair is also a critical case-pair.

Example 3. From Example 2, two maximal dispute trees can be extracted from the AA- framework, as depicted in Figure 2. The left tree in the figure is a non-admissible dispute tree which explains why the company has to pay a penalty. The dispute tree on the right in the figure is an admissible dispute tree which explains why the company does not have to pay a penalty.

[4] The original work uses $(N, ?)$ that attacks all case-pairs of which situations are not subsets of N, but, to simplify definitions in the rest of the present paper, we adapt this part of the definition following [2] instead.

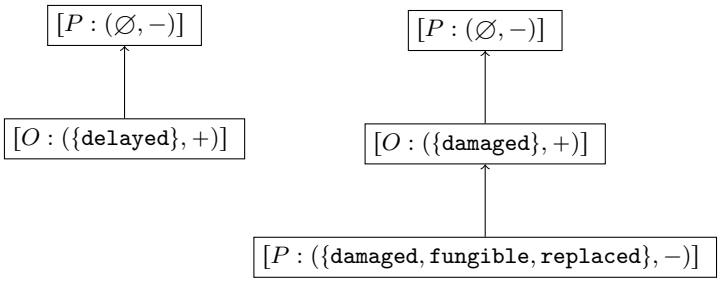


Fig. 2. Two Maximal Dispute Trees for $(\varnothing, -)$ in Example 3

3 Case Models

Case models [16] aim to formally evaluate arguments from cases. A case in case models is a logical formula, usually a conjunction of literals. A case model consists of a set of cases C, and their preference ordering $\geqslant$. The cases in a case model must be logically consistent, mutually incompatible, and different. The preference ordering $\geqslant$ in case models must be total and transitive (hence is what is called a total preorder, corresponding to a numerically representable ordering). Let $\mathcal{L}$ be a classical logical language generated from a set of propositional constants in a standard way. We write $\neg$ for negation, $\wedge$ for conjunction, $\vee$ for disjunction, $\leftrightarrow$ for equivalence, $\top$ for a tautology, and $\bot$ for a contradiction. The associated classical, deductive, monotonic consequence relation is denoted $\models$. A case model is defined as follows.

Definition 7 (Case Model [16]). *A case model is a pair $(C, \geqslant)$ with finite $C \subseteq \mathcal{L}$, such that the following hold, for all φ , ψ and $\chi \in C$:*

1. $\models \neg\varphi$ *(logically consistent, i.e. a case cannot be $\bot$);*
2. *If $\not\models \varphi \leftrightarrow \psi$, then $\models \neg(\varphi \wedge \psi)$ (mutually incompatible);*
3. *If $\models \varphi \leftrightarrow \psi$, then $\varphi = \psi$ (different);*
4. *$\varphi \geqslant \psi$ or $\psi \geqslant \varphi$ (total);*
5. *If $\varphi \geqslant \psi$ and $\psi \geqslant \chi$, then $\varphi \geqslant \chi$ (transitive).*

The strict weak order $>$ standardly associated with a total preorder $\geqslant$ is defined as $\varphi > \psi$ if and only if it is not the case that $\psi \geqslant \varphi$ (for φ and $\psi \in C$). When $\varphi > \psi$, we say that φ is (strictly) preferred to ψ. The associated equivalence relation $\sim$ is defined as $\varphi \sim \psi$ if and only if $\varphi \geqslant \psi$ and $\psi \geqslant \varphi$.

An argument in case models is a pair (φ, ψ) with φ and $\psi \in \mathcal{L}$ where φ expresses the argument's premise and ψ expresses the argument's conclusion. There are three types of argument evaluations in case models, which are coherent, presumptively valid, and conclusive (arguments which are not these three types are indeed incoherent).

Definition 8 (Argument Evaluation in Case Models [16]). *Let $(C, \geqslant)$ be a case model. Then we define, for all φ and $\psi \in \mathcal{L}$:*

- (φ, ψ) *is* coherent *with respect to* $(C, \geqslant)$ *if and only if* $\exists \omega \in C : \omega \models \varphi \wedge \psi$.
- (φ, ψ) *is* presumptively valid *with respect to* $(C, \geqslant)$ *if and only if* $\exists \omega \in C : \omega \models \varphi \wedge \psi$; *and* $\forall \omega' \in C :$ *if* $\omega' \models \varphi$, *then* $\omega \geqslant \omega'$.
- (φ, ψ) *is* conclusive *with respect to* $(C, \geqslant)$ *if and only if* $\exists \omega \in C : \omega \models \varphi \wedge \psi$; *and* $\forall \omega \in C :$ *if* $\omega \models \varphi$, *then* $\omega \models \varphi \wedge \psi$.

4 Constructing Case Models

In this section, we present our contribution of formalizing a method for constructing a case model from an AA-CBR case-base. The inspiration of the construction is from an observation of classifying possible fact-situations with respect to their unique nearest case-pair, as we demonstrate in Example 4.

Example 4. From Example 2, let the fact-domain be $\mathcal{F} = \{\texttt{delayed}, \texttt{damaged}, \texttt{fungible}, \texttt{replaced}\}$, fact-situations can be classified with their unique nearest case-pair as follows.

- $co_0 = (\varnothing, -)$ is unique nearest to:
  $\varnothing$, {fungible},{replaced},{fungible, replaced}
- $co_1 = (\{\texttt{delayed}\}, +)$ is unique nearest to:
  {delayed}, {delayed, fungible}, {delayed, replaced},
  {delayed, fungible, replaced}
- $co_2 = (\{\texttt{damaged}\}, +)$ is unique nearest to:
  {damaged}, {damaged, fungible}, {damaged, replaced}
- $co_3 = (\{\texttt{damaged}, , \texttt{fungible}, \texttt{replaced}\}, -)$ is unique nearest to:
  {damaged, , fungible, replaced}
- No unique nearest case-pair:
  {delayed, damaged}, {delayed, damaged, fungible},
  {delayed, damaged, replaced}, {delayed, damaged, fungible, replaced}

As we can see from the example, a fact-situation can monotonically grow without changing its unique nearest case-pair until it reaches *exceptional* conditions. From this observation, we define the following sets for a case-pair (X, o_x) in a case-base CB.

- $CB_{\rightarrow(X,o_x)} = \{(Y, o_y) \in CB | (Y, o_y)$ attacks (X, o_x) in the AA-framework corresponding to CB and $\mathcal{F}\}$
- $F_{\rightarrow(X,o_x)} = \bigcup_{(Y,o_y) \in CB_{\rightarrow(X,o_x)}} Y$
- $I_{\rightarrow(X,o_x)} = \{B \subseteq F_{\rightarrow(X,o_x)} | X \subseteq B \wedge \nexists (Y, o_y) \in CB_{\rightarrow(X,o_x)} Y \subseteq B\}$

We call $F_{\rightarrow(X,o_x)}$ a *boundary* of (X, o_x) and a member of $I_{\rightarrow(X,o_x)}$ an *internal sub-boundary* of (X, o_x). We have that (X, o_x) is always unique nearest to an internal sub-boundary of (X, o_x).

To define the construction, firstly, we define a naming function *name*. Let $\mathcal{N}$ be a set of propositions called a *name-domain*, distinct from $\mathcal{F}$. Each element of $\mathcal{N}$ is called a *name-proposition*. We define $name : 2^{\mathcal{F}} \times \{+, -\} \mapsto \mathcal{N}$ mapping

from every case-pair to a name proposition. For ease of exposition, we use the same symbol for referring to the case-pair and its name proposition.

Then, we define a function *case* based on an informal construction described in [16] for constructing a logical sentence from a case-pair and an internal sub-boundary of the case-pair. Let δ be a proposition called an *outcome-proposition*, which is neither a fact-proposition nor a name-proposition. The literals used for constructing the logical sentence are from five sources: (1) the outcome-proposition (2) the name-domain (3) the internal sub-boundary (4) the fact-propositions inside the boundary but outside the internal sub-boundary (5) the fact-propositions outside the boundary of the default case-pair. The *case* function is formally defined as follows.

Definition 9 (Case construction). *Let CB be a case-base with a default outcome d and a case-pair $(X, o_x) \in CB$, $B_x \in I_{\rightarrow(X,o_x)}$, and δ be an outcome-proposition. $case(X, o_x, B_x)$ is a function defined as*

$$\begin{aligned} case(X, o_x, B_x) = \quad & (o_x = d \ ? \ \delta : \neg\delta) \ \wedge \bigwedge_{n \in \mathcal{N}} (n = name(X, o_x) \ ? \ n : \neg n) \ \wedge \\ & \bigwedge_{p_i \in B_x} p_i \ \wedge (I_{\rightarrow(X,o_x)} = \{X\} \ ? \ \top : \bigwedge_{p_k \in F_{\rightarrow(X,o_x)} \setminus B_x} \neg p_k) \wedge \\ & (X = \varnothing \ ? \bigwedge_{p_l \in \mathcal{F}_{CB} \setminus F_{\rightarrow(\varnothing,d)}} p_l : \top) \end{aligned}$$

where $(a \ ? \ b : c)$ expresses a ternary conditional operator, which is interpreted as if a then b otherwise c

Since an internal sub-boundary of (X, o_x) has a unique nearest case, that is (X, o_x), *case* is a one-to-one function, namely given the logical sentence constructed from *case*, we can trace back which case-pair and which internal sub-boundary that the sentence is constructed from.

Secondly, we define a function *depth*, which is a mapping function from any case-pair to an integer, expressing the depth of attacks from the default case-pair to the considered case-pair. This function is used for determining the preference between cases as follows.

Definition 10 (Attack depth). *Let CB be a case-base with a default outcome d and a critical case-pair (X, o_x), and $\langle CB, attacks \rangle$ be the AA-framework corresponding to CB and $\mathcal{F}$. $depth(X, o_x)$ is a function defined as*

$$depth(X, o_x) = \begin{cases} 0 & \text{if } X = \varnothing \\ 1 + \max\limits_{(X,o_x) \ attacks \ (Y,o_y)} depth(Y, o_y) & \text{otherwise} \end{cases}$$

Using these two functions, we present the following formal method for constructing case models as follows.

Definition 11. *Let CB be a case-base with a default outcome d. We say a case model $(C, \geqslant)$ is constructed from CB if and only if the following conditions hold.*

1. *for every critical case-pair $(X, o_x) \in CB$ and $B_x \in I_{\rightarrow(X,o_x)}$, there exists $case(X, o_x, B_x) \in C$; and*
2. *for every critical $(X, o_x), (Y, o_y) \in CB$, $B_x \in I_{\rightarrow(X,o_x)}$, and $B_y \in I_{\rightarrow(Y,o_y)}$ such that $c_1 = case(X, o_x, B_x), c_2 = case(Y, o_y, B_y) \in C$, $c_1 \geqslant c_2$ if and only if $depth(X, o_x) \leqslant depth(Y, o_y)$; and*
3. *there are no other cases in C except those given by 1.*

Since *case* is a one-to-one function, cases in a constructed case model are different from each other. With the layout of negations in the construction, cases in a constructed case model are mutually incompatible. The preference ordering is total and transitive since it is derived from numeric comparisons.

Table 1. Constructing cases in case model from the working example

case-pairs and boundaries	Internal sub-boundary	Cases in case model
$co_0 = (\varnothing, -)$ Boundary = {`delayed`, `damaged`}	$\varnothing$	$c_0 : \delta \wedge co_0 \wedge \neg co_1 \wedge \neg co_2 \wedge \neg co_3$ $\wedge$ `fungible` $\wedge$ `replaced`
$co_1 = (\{$`delayed`$\}, +)$ Boundary = {`delayed`}	{`delayed`}	$c_{1a} : \neg\delta \wedge \neg co_0 \wedge co_1 \wedge \neg co_2 \wedge \neg co_3$ $\wedge$ `delayed`
$co_2 =$ $(\{$`damaged`$\}, +)$ Boundary = {`damaged`, `fungible`, `replaced`}	{`damaged`} {`damaged`, `fungible`} {`damaged`, `replaced`}	$c_{1b} : \neg\delta \wedge \neg co_0 \wedge \neg co_1 \wedge co_2 \wedge \neg co_3$ $\wedge$ `damaged` $\wedge \neg$`fungible` $\wedge \neg$`replaced` $c_{1c} : \neg\delta \wedge \neg co_0 \wedge \neg co_1 \wedge co_2 \wedge \neg co_3$ $\wedge$ `damaged` $\wedge$ `fungible` $\wedge \neg$`replaced` $c_{1d} : \neg\delta \wedge \neg co_0 \wedge \neg co_1 \wedge co_2 \wedge \neg co_3$ $\wedge$ `damaged` $\wedge \neg$`fungible` $\wedge$ `replaced`
$co_3 = (\{$`damaged`, `fungible`, `replaced`$\}, -)$ Boundary ={`damaged`, `fungible`, `replaced`}	{`damaged`, `fungible`, `replaced`}	$c_2 : \delta \wedge \neg co_0 \wedge \neg co_1 \wedge \neg co_2 \wedge co_3$ $\wedge$ `damaged` $\wedge$ `fungible` $\wedge$ `replaced`

The preference ordering: $c_0 > c_{1a} \sim c_{1b} \sim c_{1c} \sim c_{1d} > c_2$

From Example 1, a case model $(C, \geqslant)$ is constructed as in Table 1. Case c_0, which is a most preferred case in C, is constructed from the default case-pair co_0. `fungible` and `replaced` are attached to the case since they are not in the boundary of the default case-pair. c_{1a} is constructed from co_1 since it has only one internal sub-boundary. In contrast, c_{1b}, c_{1c}, c_{1d} are constructed from the same case-pair co_2 since it has three internal sub-boundaries. $c_{1a}, c_{1b}, c_{1c}, c_{1d}$ are immediately less preferred than c_0 because they are constructed from the case-pairs that directly attack the default one. Meanwhile, c_2 is constructed from co_3 and c_2 is the least preferred in C.

5 Explaining Case Models

In this section, we present another contribution of developing dispute trees for explaining case models. To develop the explanation, we first look into the concept of analogy, which is defined as follows [16].

Definition 12 (Analogy). *Let $\mathcal{L}$ be a classical logical language, $(C, \geqslant)$ be a case model, and $\sigma \in \mathcal{L}$ be a situation. We say $\alpha \in \mathcal{L}$ expresses an* analogy *of a case $\omega \in C$ and σ if $\omega \models \alpha$ and $\sigma \models \alpha$.*

For any case ω and any situation σ, we have that $\top$ is the most general analogy of ω and σ, and $\omega \vee \sigma$ is the most specific analogy of ω and σ [19]. By extending the concept of specificity from AA-CBR, we introduce a *literal analogy* as an analogy in the form of $\top$ or a conjunction of literals. This makes $\top$ still the most general literal analogy of ω and σ, but $\omega \vee \sigma$ is not always the most specific literal analogy due to the logical or. The exception is that sometimes there is a conjunction of literals that is equivalent to $\omega \vee \sigma$, in that case, such a conjunction is the most specific literal analogy.

Definition 13 (Literal Analogy). *We say an analogy α is a* literal analogy *of ω and σ if and only if α is $\top$ or a conjunction of literals. and we say a literal analogy α is the* most specific literal analogy *of ω and σ if and only if for every literal analogy α' of ω and σ, $\alpha \models \alpha'$.*

By the concept of literal analogy, we introduce a new type of rebuttals called *specificity rebuttal*, based on the attack relations in AA-CBR, also inspired by [11, 14]. It intuitively means the rebuttal consists in finding a more specific literal analogy from a most preferred case with the opposite outcome.

Definition 14 (Specificity Rebuttal). *Let $\mathcal{L}$ be a classical logical language, $(C, \geqslant)$ be a case model, (φ, ψ) be a presumptively valid argument, and $\sigma \in \mathcal{L}$ be a situation. We say a non-tautologous $\chi \in \mathcal{L}$ (i.e. $\chi \neq \top$) is* specificity rebutting *the argument with respect to σ if and only if*

- *$\exists \omega \in C : \omega \models \varphi \wedge \neg\psi$; $\forall \omega' \in C$: if $\omega' \models \varphi \wedge \neg\psi$, then $\omega \geqslant \omega'$ (ω is a most preferred case in the set of such ω' with respect to $\geqslant$); and*
- *$\varphi \wedge \chi$ is a most specific literal analogy of ω and σ.*

Now, we present dispute trees in case models based on those in AA-CBR as follows.

Definition 15 (Dispute Tree in Case Models). *Let $\sigma \in \mathcal{L}$ be a situation, $(C, \geqslant)$ be a case model, and ψ_0 be a logic formula such that $(\top, \psi_0)$ is presumptively valid with respect to $(C, \geqslant)$. A* dispute tree *for ψ_0 with respect to $(C, \geqslant)$ and σ is a tree $\mathcal{T}$ such that:*

1. *every node of $\mathcal{T}$ is of the form $[L : (\varphi, \psi)]$ where $L \in \{P, O\}$ and $\varphi, \psi \in \mathcal{L}$.*
2. *the root of $\mathcal{T}$ is $[P : (\top, \psi_0)]$*

3. *for every* $[P : (\varphi, \psi)]$ *and for every* $\chi \in \mathcal{L}$ *that is specificity rebutting* (φ, ψ) *with respect to* σ*, there exists* $[O : (\varphi \wedge \chi, \neg\psi)]$ *as a child of* $[P : (\varphi, \psi)]$*;*
4. *for every* $[O : (\varphi, \psi)]$*, there exists at most one child* $[O : (\varphi \wedge \chi, \neg\psi)]$ *such that* χ *is specificity rebutting* (φ, ψ) *with respect to* σ*;*
5. *there are no other nodes in* $\mathcal{T}$ *except those given by 1-4.*

A dispute tree $\mathcal{T}$ *is a* maximal *dispute tree if and only if for every* $[O : (\varphi, \psi)]$ *which is a leave in* $\mathcal{T}$*, no* $\chi \in \mathcal{L}$ *that is specificity rebutting* (φ, ψ) *with respect to* σ*.*

We prove a theorem that a maximal dispute tree in the constructed case models is homomorphic to some maximal dispute tree in AA-CBR, i.e. there is a mapping (not always bijective) from nodes in a maximal dispute tree in the constructed case models to nodes in the corresponding maximal dispute tree in AA-CBR such that the parent-child adjacencies are still preserved. Roughly speaking, a maximal dispute tree in the constructed case models can be reduced into a maximal dispute tree in AA-CBR.

Theorem 1. *Given a fact-situation* N*, a case-base* CB *with a default case-pair* $(\varnothing, d)$*; the corresponding AA-framework* $\langle AR, attacks\rangle$*; and the case model* $(C, \geqslant)$ *constructed from* CB *with respect to a proposition* δ*. A maximal dispute tree* $\mathcal{T}$ *for* δ *with respect to* $(C, \geqslant)$ *and* $\bigwedge_{p_i \in N} p_i$ *is homomorphic to some maximal dispute tree* $\mathcal{T}'$ *for* $(\varnothing, d)$ *with respect to* $\langle CB, attacks\rangle$*, with a homomorphic mapping from a node* $[L : (\varphi, \psi)]$ *in* $\mathcal{T}$ *to a node* $[L : (X, o)]$ *in* $\mathcal{T}'$ *such that a most preferred case in* $\{\omega | \omega \models \varphi \wedge \psi\}$ *is constructed from* (X, o)*.*

Proof. We prove by induction that $\mathcal{T}$ is homomorphic to some maximal dispute tree $\mathcal{T}'$ for $(\varnothing, d)$ with respect to $\langle AR, attacks\rangle$.

- **base case:** The root $[P : (\top, \delta)]$ of $\mathcal{T}$ corresponds to the root $[P : (\varnothing, d)]$ of $\mathcal{T}'$ $(\top, \delta)$ has grounding in a most preferred case in $\{\omega | \omega \models \delta\}$ with respect to $\geqslant$, which is always constructed from $(\varnothing, d)$.
- **inductive step:** If $[O : (\varphi', \neg\psi)]$ is a child of $[P : (\varphi, \psi)]$ in $\mathcal{T}$ and $[P : (\varphi, \psi)]$ corresponds to $[P : (Y, o_y)]$ in $\mathcal{T}'$, then there exists a most preferred case ω_x in the set $\{\omega | \omega \models \varphi \wedge \neg\psi\}$ with respect to $\geqslant$. Since ω_x is constructed from some $(X, o_x) \in CB$, we have that (X, o_x) attacks (Y, o_y) because $o_x \neq o_y$ (as $\omega_x \models \neg\psi$); $Y \subsetneq X$ (as there exists a non-tautologous χ such that $\omega_x \models \varphi \wedge \chi$); and $\nexists (Z, o_x) \in AR$ with $Y \subsetneq Z \subsetneq X$ (as ω_x is a most preferred case in the set, hence (X, o_x) is far from (Y, o_y) by a distance of *attacks* 1). Therefore, $[O : (X, o_x)]$ is a child of $[P : (Y, o_y)]$ (This can be applied analogously for a case that $[P : (\varphi', \neg\psi)]$ is a child of $[O : (\varphi, \psi)]$).
- If $\mathcal{T}$ is maximal, then for all opponent node $[O : (\varphi, \psi)]$ which are leaves in $\mathcal{T}$, no χ is specificity rebutting (φ, ψ) with respect to $\bigwedge_{p_i \in N} p_i$. Hence, there is no $(X, o_x) \in AR$ that attacks (Y, o_y) if $[O : (Y, o_y)]$ corresponds to $[O : (\varphi, \psi)]$, otherwise there is $\chi = \bigwedge_{p_j \in X \setminus Y} p_j$ that is specificity rebutting (φ, ψ) with respect to $\bigwedge_{p_i \in N} p_i$, which leads to the contradiction. Hence, $\mathcal{T}'$ is maximal.

Figure 3 shows examples of maximal dispute trees for δ with respect to the case model in Table 1 and the situation $\texttt{delayed} \wedge \texttt{damaged} \wedge \texttt{fungible} \wedge \texttt{replaced}$. We have that the dispute trees on the left and the right of the Figure 3 are homomorphic to the dispute trees on the left and the right of Figure 2 respectively.

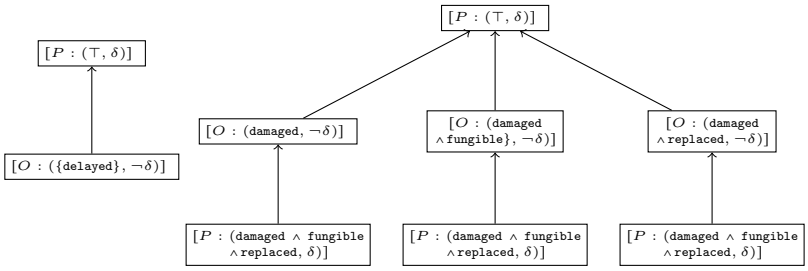


Fig. 3. Examples of maximal dispute trees for δ with respect to the case model constructed from the example

6 Discussion

In this paper, we present a method for constructing case models and dispute trees for explaining case models based on AA-CBR. However, unlike dispute trees in AA-CBR [6, 7] that start with a default case, dispute trees in case models can start with any arbitrary formula ψ such that $(\top, \psi)$ is presumptively valid. Although the formula is originally derived from a proposition representing a default outcome, it is not necessary to be such a proposition. Since previous studies [2, 13] show that AA-CBR case-bases can be translated into stratified logic programs, it follows immediately from this paper that case models constructed from AA-CBR case-bases can also be translated into stratified logic programs. Unfortunately, not every case models can be translated into stratified logic programs because case models can express inconsistencies, which stratified logic programs cannot express. Future research could investigate whether there is a programming paradigm or a logical framework that every case model can be translated into. Interesting candidates are answer set programming or defeasible logic since they can express inconsistencies.

Besides AA-CBR, it is interesting to investigate constructing and explaining case models from other perspectives, such as from precedential constraint. Some differences between dispute trees in case models and dialogue games in precedential constraint [12, 18] are, for example, dispute trees in case models play on hypothetical arguments, i.e. arguments that might not have grounding in real precedent cases, while dialogue games in precedential constraint play on real precedent cases. Another difference is that the dispute trees in case models studied here consider only specificity rebuttals. They do not consider the idea

in precedential constraint that a precedent case can defend a decision for a new case with stronger support without using specificity. Therefore, new types of explanations and attack relations in case models might be found if we construct and explain case models from precedential constraint or other perspectives.

7 Conclusion

This paper presents a method of constructing case models based on an abstract argumentation for case-based reasoning (AA-CBR). The constructed case models consists of cases, each of which is constructed from each internal sub-boundary of each critical case-pair in the case-base, and preferences over cases, which are determined by the distance of attacks between the default case-pair and the considering case-pair in the corresponding argumentation framework in AA-CBR. By connecting AA-CBR to case models, we can derive dispute trees with respect to a case model constructed and a situation. It has been shown that the maximal dispute trees in case models can be reduced into maximal dispute trees in AA-CBR. In future work, it would be interesting to study constructing and explaining case models from other perspectives and to study relations between case models and other programming paradigms or logical frameworks.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Numbers, JP17H06103 and JP19H05470 and JST, AIP Trilateral AI Research, Grant Number JPMJCR20G4.

References

1. Aleven, V.: Teaching case-based argumentation through a model and examples. Ph.D. thesis, University of Pittsburgh (1997)
2. Athakravi, D., Satoh, K., Law, M., Broda, K., Russo, A.: Automated inference of rules with exception from past legal cases using ASP. In: International Conference on Logic Programming and Nonmonotonic Reasoning. pp. 83–96. Springer International Publishing, Cham (2015)
3. Atkinson, K., Bench-Capon, T., Bollegala, D.: Explanation in AI and law: Past, present and future. Artificial Intelligence **289**, 103387 (2020)
4. Bench-Capon, T.J.: Hypo's legacy: introduction to the virtual special issue. Artificial Intelligence and Law **25**(2), 205–250 (2017)
5. Cocarascu, O., Stylianou, A., Čyras, K., Toni, F.: Data-empowered argumentation for dialectically explainable predictions. In: ECAI 2020, pp. 2449–2456. IOS Press, Amsterdam, The Netherlands (2020)
6. Cyras, K., Satoh, K., Toni, F.: Abstract argumentation for case-based reasoning. In: Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning. pp. 243–254. AAAI Press, CA, USA (2016)
7. Cyras, K., Satoh, K., Toni, F.: Explanation for case-based reasoning via abstract argumentation. In: Computational Models of Argument. pp. 243–254. IOS Press, Amsterdam, The Netherlands (2016)

8. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial intelligence **77**(2), 321–357 (1995)
9. Henderson, J., Bench-Capon, T.: Describing the development of case law. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. pp. 32–41. ICAIL '19, Association for Computing Machinery, New York, NY, USA (2019)
10. van Leeuwen, L., Verheij, B.: A comparison of two hybrid methods for analyzing evidential reasoning. In: Legal Knowledge and Information Systems. pp. 53–62. IOS Press, Amsterdam, The Netherlands (2019)
11. Prakken, H.: A tool in modelling disagreement in law: preferring the most specific argument. In: Proceedings of the 3rd international conference on Artificial intelligence and law. pp. 165–174. Association for Computing Machinery, New York, NY, USA (1991)
12. Prakken, H., Ratsma, R.: A top-level model of case-based argumentation for explanation: formalisation and experiments. Argument & Computation **13**(2), 159–194 (2022)
13. Satoh, K., Kubota, M., Nishigai, Y., Takano, C.: Translating the Japanese Presupposed Ultimate Fact Theory into logic programming. In: Proceedings of the 2009 Conference on Legal Knowledge and Information Systems: JURIX 2009: The Twenty-Second Annual Conference. pp. 162–171. IOS Press, Amsterdam, The Netherlands (2009)
14. Simari, G.R., Loui, R.P.: A mathematical treatment of defeasible reasoning and its implementation. Artificial intelligence **53**(2-3), 125–157 (1992)
15. Verheij, B.: Formalizing value-guided argumentation for ethical systems design. Artificial Intelligence and Law **24**(4), 387–407 (2016)
16. Verheij, B.: Formalizing arguments, rules and cases. In: Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law. p. 199–208. ICAIL '17, Association for Computing Machinery, New York, NY, USA (2017)
17. Verheij, B.: Proof with and without probabilities: Correct evidential reasoning with presumptive arguments, coherent hypotheses and degrees of uncertainty. Artificial Intelligence and Law **25**, 127–154 (2017)
18. van Woerkom, W., Grossi, D., Prakken, H., Verheij, B., Čyras, K., Kampik, T., Cocarascu, O., Rago, A., et al.: Justification in case-based reasoning. In: Proceedings of the First International Workshop on Argumentation for eXplainable AI. pp. 1–13. CEUR Workshop Proceedings, Utrecht University, The Netherlands (2022)
19. Zheng, H., Grossi, D., Verheij, B.: Logical comparison of cases. In: AI Approaches to the Complexity of Legal Systems XI-XII, pp. 125–140. Springer, Cham (2020)

Modeling the Judgments of Civil Cases of Support for the Elderly at the District Courts in Taiwan

Chao-Lin Liu†, Wei-Zhi Liu↑, Po-Hsien Wu↟, Sieh-chuen Huang§, Ho-Chien Huang‡

†↑↟ Department of Computer Science
‡College of Law
National Chengchi University, Taiwan

§ College of Law
National Taiwan University, Taiwan

{†chaolin, ↑109753157, ↟111753120, ‡110601014}@nccu.edu.tw, §schhuang@ntu.edu.tw

Abstract. The problem of legal judgment prediction (LJP) has attracted strong attention of researchers of both the law and computer science communities in recent years. The majority of the previous LJP work is for the criminal cases. We report our attempt to predict the judgments for the cases about the support for the elderly, which is an instance of the civil cases. We investigated the effects of choosing different design parameters in our decision models, which adopted the concepts of both traditional machine learning and deep learning. The results of the empirical evaluations showed some encouraging results, but some others uncovered the challenges that remain to be tackled.

Keywords: Legal Informatics, Artificial Intelligence, Machine Learning, Deep Learning, Modeling Court Decisions, Legal Judgment Prediction.

1 Introduction

With the accelerated advancement and increasing affordability of the computing technology, the technology has been applied to a wide range of areas, including legal informatics. Yamakoshi et al. discussed the translation of Japanese law articles for foreign readers [16], Rabelo et al. reported their studies on text entailment between legal statements [13], and Komamizu et al. shared their experience in identifying relevant parts between legal documents [5] in recent JURISIN workshops.

The work reported in this article is an instance of legal judgment prediction (**LJP**) [2]. Due to the recent expediated advancement in artificial intelligence and machine learning, the problem of predicting judges' judgments based on the information about the criminal activities or legal disputes has attracted attention of more and more legal experts and computer science researchers [1].

The majority of the current research activities in LJP is about the criminal cases, partially because specific criminal activities must be reported to establish the conditions of starting a criminal case. If one may identify the criminal activities sufficiently and explicitly, then it is possible for a computer system to recommend the types of sentences and even the lengths of imprisonment for some criminal cases, based on the results of the statistical analysis of the previous cases. The Judicial Yuan of Taiwan provides a

legal judgment recommender system for eight types of criminal crimes on line.[1] The main users of such a public service might not be the human judges. Such a recommender is both educational and helpful for ordinary civilians and lawyers.

Some researchers have started to explore the domain of civil cases. It is relatively harder to interpret and understand the intentions and functions of statements in civil cases, and researchers are working on more fundamental issues. Wu et al. attempted to predict courts' views based on plaintiffs' claims [15]. Liu et al. aimed at identifying and differentiate the statements of the plaintiffs and of the defendants in judgment documents [9]. Zhao et al. overviewed the sub tasks for the predictions of civil cases [17].

We report results of our modeling the court decisions for cases of the support for the elderly. Muhlenbach et al. worked on the alimony issues for divorce cases [12]. Their interests included the jurisdictional factors and the economic consequences. They applied tree-based methods for classification and regression models for predicting the amount of alimony. The structures of our problems are similar, but we report our experience of applying technically more complex models in this paper.

We discuss the data source in Section 2, and define our goals in Section 3. In Section 4, we present the preliminary results of the predictions of whether the judges would accept or dismiss the plaintiffs' requests. In Sections 5 and 6, we delineate the applications of a model-tree model for estimating the granted amount of fund. In Section 7, we offer short concluding remarks.

2 Data Source

2.1 Open Data of Taiwan Judicial Yuan (TWJY)

The Judicial Yuan is the highest official agency for governing the judicial system in Taiwan. By law, the judgment documents of the courts should be published, except those that are not allowed to be published by law. The website for the open documents (TWJY, henceforth) is in Chinese, and is publicly accessible on the Internet.[2]

TWJY contains judgment documents for lawsuits since January 1996, and has accumulated about 18 million documents. In the first four years, the available data were limited, and mostly were from special courts. Since 2000, the documents of the three layers of courts, i.e., district, high, supreme courts, started to become available. The Judicial Yuan updates the contents of TWJY monthly, with a three-month lag. Namely, the judgment documents of January will not be published until April. New judgments documents from different types and levels of courts will be published and can be downloaded as a compressed file. Sometimes, the published documents may be retracted due to legal reasons, so the number of available documents does not remain extremely stable. The published documents are anonymized according to the law as well, and it is the government's responsibility to protect privacy for the individuals involved in the lawsuits.

[1] 量刑趨勢建議系統 (Legal Judgment Recommender): https://sen.judicial.gov.tw/pub_platform/sugg/index.html (in Chinese), last accessed 2023/05/21

[2] Open data of Taiwan Judicial Yuan: https://opendata.judicial.gov.tw/, last accessed 2023/05/21

2.2 Selecting Relevant Documents

Although there are about 18 million judgment documents, a typical research usually focuses on one or some special categories of lawsuits, and does not need to use all those documents. In our current work, we focus on the issues regarding the support for the elderly, which is related to family support problems and belongs to the category of civil cases. To this end, we select the judgment documents that meet a special set of criteria.

Each of the judgment documents in TWJY is a JSON file, and follows a top-level structure, including the long identification number (JID), the year when the lawsuit started in terms of the Taiwan calendar (JYEAR), the abbreviated code for the type of the lawsuit (JCASE), the short identification number for the lawsuit (JNO), the date for the judgment in terms of the Western calendar (JDATE), the category of the lawsuit (JTITLE), and the full text for the judgment document (JFULL).

To find specific type of judgments from TWJY, we can select the documents based on whether or not the JCASE and the JTITLE of the documents contain specific keywords. As a preliminary step, we extracted documents whose JTITLE has the word "扶養費" (amount of fund for support), while ignoring documents whose JTITLE is either "請求扶養費代墊" or "請求扶養費代墊款". These two sub-types are for requesting the returning of the fund that the plaintiffs had paid for the defendants, so are not related to our research goals. The contents of the JCASE may indicate more subtleties about the judgments. If the JCASE contains the characters in {審, 調, 補, 促, 抗, 消, 更, 上, 續, 救, 他, 高等, 最高, 婚} [8], they are not directly about the judgments about whether to grant money or the amount of granted fund to the plaintiffs at the district courts. It is not easy to explain each of these legal exclusions in this paper. By adopting this list, we can exclude cases for appeals to the high courts and the supreme court, for instance. In this study, we want to focus on the judgments of the district courts.

Furthermore, we dropped documents that passed the aforementioned filtering steps, if they were about transferring the jurisdiction from a court to another. This could also be achieved by detecting keyworks like "本件移送" in the JFULL.

For the period between January 2000 and December 2021, we identified only 1930 judgment documents from TWJY that satisfied all of these criteria. The judges granted the plaintiffs a certain amount of fund in 938 cases, and dismissed 992 cases.

2.3 A Relevant Social Factor and Mediation

It might be surprising that we obtained only 1930 documents from TWJY, which claims to have about 18 million documents in total. We can offer two factors for this phenomenon.

The documents included in TWJY are for all administrative, criminal, and civil courts in Taiwan for the time period between 1996 to the present. Among them, the cases of support for the elderly is not the most frequent cases.

More specifically, Taiwan, as an Asian country, it is not common for family members to bring family issues to the courts. In the old days, family problems should be resolved within the family, and should not even let any outsiders know. Otherwise, that would become a shame for the whole family.

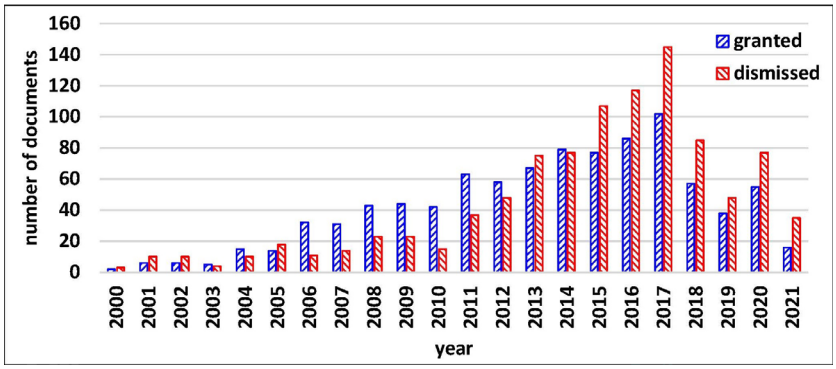


Fig. 1. The distribution of the selected judgment documents for the cases of support for the elderly between 2000 and 2021

In addition, the courts will offer mediation between the plaintiffs and the defendants. Hence, not all of the litigations would result in the final court decisions. Therefore, the number of judgment documents that we could use may not be as large as one may have expected. In fact, Huang, a professor in Law, also reported a similar amount of judgment documents in an unpublished conference presentation [4].

Table 1. Regular expressions for extracting claims of the plaintiffs and of the defendants

	regular expressions
plaintiffs	(聲請|聲請人|原告).{0,10}(主張|略以|意旨)
defendants	(相對人|被告).{0,10}(主張|略以|意旨|則以|抗辯)

We show the distributions of the number of the selected cases over the years, including "granted" and "dismissed", in Figure 1. We can observe the increasing trends, but the total number of cases remains low.

3 Problem Definitions and Data Preprocessing

3.1 Judgment and Grant Predictions

Given a selected document, there are two tasks we can do. The first one is to predict whether the plaintiffs' requests can be granted, and the second is to predict the amount of granted monthly amount for support.

We treat the first task as a classification problem, as we have indicated in Figure 1. We need to extract the claims of the plaintiffs and of the defendants from the JFULL field of the documents, and must not use any other parts that may shed light on the final judgments, e.g., paragraphs that mentioned the cited law articles may indicate the opinions of the judges and so the final decisions. This can be achieved by observing the regularities of the statements of how the documents recorded the claims of the plaintiffs and of the defendants. Table 1 shows the regular expressions (mixed with Chinese words) that can catch the regularities. Based on the extracted statements, we trained classifiers to categorize the cases into "**granted**" and "**dismissed**" categories.

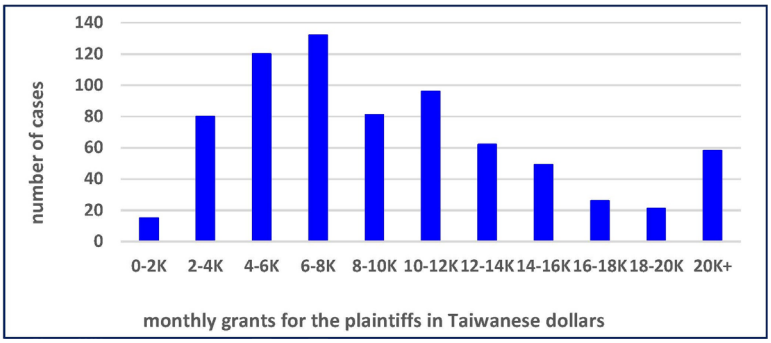


Fig. 2. The distributions of monthly grants to the plaintiffs

We treat the second task as a regression problem. A relatively more common approach for the LJP problems is to discretize the range of the penalties, and treat the LJP problems as a classification task. We would identify the features for building the regression models with both manual and algorithmic methods, and details will be provided in Section 5.1. We show the distribution of the monthly grants to plaintiffs who won their cases in Figure 2. The amounts of the granted fund for support are not large, partially due to the fact that there are social programs that could support the needed in Taiwan. The range of the distribution is not small, so achieving precise prediction is not a trivial goal. We will discuss more details in Sections 5 and 6.

3.2 Preprocessing: Word Segmentation and the Blurring Procedure

Due to the size of our data, we would like to compare the effectiveness of using traditional machine learning methods and of using deep learning models. Hence, we need to do word segmentation for Chinese strings when building some classifiers. We adopted the CKIP classifier that was maintained by the Academia Sinica.[3]

The Judicial Yuan anonymized the documents in TWJY by replacing the given names by circles. For instance, a person name in Chinese "劉昭麟" will be substituted by "劉○○". In our current study, we will replace an anonymized name with "某人" ("somebody"). This is because we do not think it is necessary to consider the surnames of the plaintiffs or of the defendants in our classification models. Similarly, we may replace the street names by "somewhere" ("某地") and time expressions by "a point of time" ("某時"), unless it is necessary to consider the locations and the information about time in the lawsuit (to be explained in Section 5). This **blurring** procedure can be achieved by regular expressions and sometimes by small functions that consider the contexts. Here is a concrete example of effects of the blurring step: changing "被告丁○○答辯略以：㈠92年間被告丁○○在中壢市上班" into "被告某人答辯略以：㈠某時間被告某人在某地上班".

We may identify person names, place names, and time expressions with two possible methods. When we use the CKIP to do Chinese word segmentation, we will also obtain information about the part-of-speech (POS) of the words. The words about person

[3] CKIP: https://github.com/ckiplab/ckiptagger, last accessed 2023/05/21

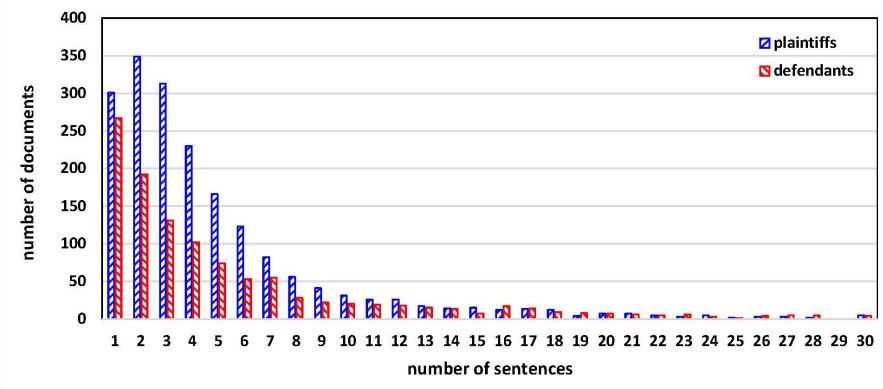


Fig. 3. The distributions of the number of sentences in the claims of the plaintiffs and the defendants

names, locations, and time expressions have specific POS labels. We can also employ tools for named-entity recognition (written as **NER** in the literature) for this task.

4 The Grant and Dismiss Decisions

4.1 Vectorizations of the Texts: TF-IDF and Sentence-BERT

We vectorized the claims of the plaintiffs and of the defendants with two methods: (1) TF-IDF [11] and (2) with the help of the Sentence-BERT (**SBERT**, henceforth) [14]. TF-IDF is a well-known method for vectorizing text, and there are many variants. We used the CKIP to segment the Chinese strings and relied on the tool **TfidfVectorizer** in the **scikit learn**[4] to build the TF-IDF models for classifying individual judgments as "granted" or "dismissed".

We also relied on SBERT to provide BERT-style sentence vectors for the claims of the plaintiffs and of the defendants. Since there are no strict standards for splitting Chinese statements in "sentences", we chose to split Chinese strings by four punctuation marks: "。！？；". After this sentence splitting step, we can find the distributions of the number of sentences in the claims of the plaintiffs and of the defendants. Figure 3 shows the distribution of the number of sentences (horizontal axis) in the claims of the plaintiffs and of the defendants. Due to the width of page, we do not show the complete distribution. The claims can have as many as 50 sentences. On average, there are six and seven sentences in the claims of the plaintiffs and of the defendants, respectively.

After splitting the sentences, we could calculate the number of Chinese characters in the sentences in the claims of the plaintiffs and of the defendants, and observed their distributions, which is analogous to how we created Figure 3. We could not include the chart due to page limits for the JURISIN submissions. The average number of characters in the sentences of the plaintiffs and of the defendants are 88 and 83, respectively. Less than 0.2% of the sentences have more than 512 characters that a typical BERT model will accept. Hence, we would do more padding than truncation. We have to feed

[4] scikit learn: https://scikit-learn.org/stable/, last accessed 2023/05/21

the same number of sentences for each plaintiff and the same number of sentences for the defendants to the classifiers, as shown in Figure 4, and we chose to use the average numbers of sentences in the claims that we reported above.

4.2 Classification of the Judgment Documents

Recall that we had only 1930 documents. Nevertheless, we still had to split these documents for training and testing at the 8:2 ratio. Then, 20% of the training data would be used for validation. We fed the TF-IDF vectors to a naïve Bayes (**NB**) model and a logistic regression (**LR**) model, and we tried two different flows with the SBERT, shown in Figures 4 and 5. The numbers in the parentheses in the rounded boxes were the numbers of output units. In Figure 5, the BiLSTM symbol denotes two BiLSTM layers.

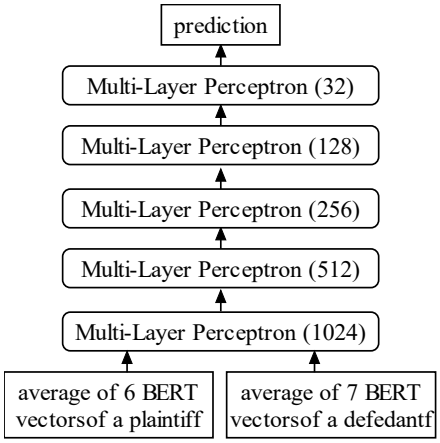


Fig. 4. A model for the SBERT experiment

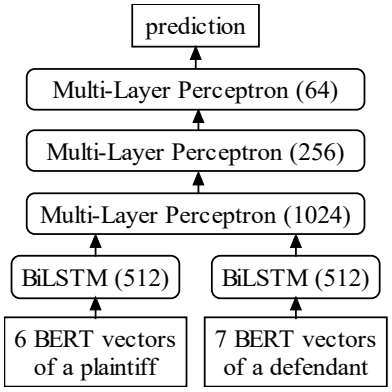


Fig. 5. Another model for the SBERT

On top of these four combinations, we tried to compare how whether we did the blurring procedure or not would influence the classification results. When we conducted the blurring procedure, the resulting vocabulary size in the TF-IDF model was about 18000. If not, the vocabulary size increased to about 24000. In both cases, stop words were ignored, but we had not tried to reduce the dimensionality by SVD yet.

When we trained these models in Figures 4 and 5, we chose Adam as the optimizer, set the initial learning rate to 0.002, used the binary cross entropy as the loss function, and let the batch size be 128. Training would stop if the loss for the validation data did not improve for two consecutive epochs. Moreover, we repeated each of these experiments 100 times by re-splitting the data to gather the statistics about the observed distributions of the accuracy and F_1 measure.

Table 2. The averages of the performance measures of the eight classifiers

vectorizer+model	blur	accuracy	F_1
TF-IDF+NB	no	0.5603	0.5593
TF-IDF+NB	yes	0.5709	0.5695
TF-IDF+LR	no	0.7057	0.7070
TF-IDF+LR	**yes**	**0.7147**	**0.7152**
Fig. 4	no	0.5806	0.5601
Fig. 4	yes	0.5907	0.5695
Fig. 5	no	0.5994	0.5204
Fig. 5	yes	0.5998	0.5257

Table 2 shows the results. The "vectorizer+model" column summarizes the methods for vectorization and the classification models. Doing the blurring step led to better

performance than not constantly. It might be surprising that using TF-IDF with a logistic regression (LR) performed the best. We note that we have not fine-tuned the SBERT due to insufficient data. This is also one of the possible reasons why we observed that the F_1 measures, in the boxplots in Figure 6, that were achieved by the neural network models dispersed relatively wider in the 100 experiments. At the time of writing, we have figured out where to find the extra data to fine tune the SBERT, and will do so soon. In fact, we have also published some preliminary results for another approach that considers the semantic functions of statements in the judgment documents [9], and the results of some extended experiments have validated the potentials of using the pretrained BERT models.

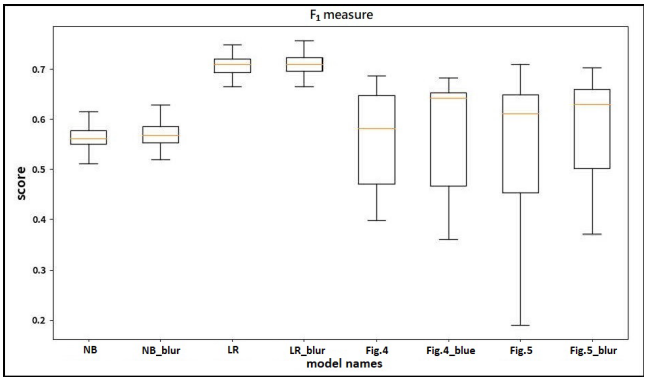


Fig. 6. The boxplots for the F_1 measures for the eight classifiers [10]

5 A Model-Tree Approach for Predicting the Grants

If the judges do not dismiss the plaintiffs' requests, the judges will then determine the amount of grants. There are two types of common decisions. The grants may be stated in an annual or a monthly amount. For building our models, we would convert all the grants to a monthly scale. There might be multiple defendants in an individual lawsuit, e.g., the plaintiffs had more than one child. In such cases, the defendants would have their shares to pay the grant to the plaintiffs, and the shares might be different. In such cases, we would sum up the shares of the defendants, and used the total as the grant for the lawsuit. Our goal was to predict the total monthly amounts that were granted to the plaintiffs, whose distribution was depicted in Figure 2.

We do not aim at predicting all types of the solutions for the support for the elderly. Predicting the shares of individual defendants is a challenging and interesting goal. In some cases, the court will grant a one-time lump sum to the plaintiffs because the life expectance is short. In other cases, the lawsuits might be interrupted due to the success of the mediation process. We may try to tackle these special topics in the future.

As we wish to predict the grants from the perspective of regression, rather than classification, we need to identify the factors that may influence the amount of the grant. At this stage, we rely on legal knowledge, e.g., [4,7], and our observations from reading the documents to select the factors.

5.1 Factors Influencing the Grants

The factors which may influence the amounts of the grants may be categorized into two main sources: financial and social factors.

It is easy to understand that the **cost-of-living index** is an important factor. This is one example when we need to know the exact places where the plaintiffs live. In Taiwan, the needed, including the qualified elderly, may receive **social allowance** from the central and the local governments. In a lawsuit, the plaintiffs must **request an amount of grant**, which may include their **special needs**. By law, there may be a certain number of persons who should be responsible for the supports of the plaintiff(s), and, among these groups, some or all are listed as the defendants. Notice that it is possible that not all of the **responsible persons** are to be included in the defendants. Knowing this key difference, we try to find information about the total monthly incomes and the total (estimated) values of the estate of the responsible persons and of the defendants that are recorded in the judgment documents. Table 3 summarizes these items.

Table 3. Financial factors

code	summary
C	cost-of-living index
G	social allowance
O	special needs
A	requested amount
P	number of defendants
N	number of responsible persons
I	total monthly income of P
E	total estate of P
NI	total monthly income of N
NE	total estate of N

Table 4. Social factors

code	summary
S1	plaintiffs have bad records?
S2	defendants domestically abuse?
S3	plaintiffs domestically abuse?
S4	plaintiffs responsible?
S5	defendants disabled?
S6	monthly incomes above median?
S7	monthly incomes below minimum salary?
RI	(I/P)/(NI/N)
RE	(E/P)/(NE/N)

According to [7] and our own observations, the courts will consider some behavioral factors of the litigants. These factors include (1) whether the plaintiffs have some bad records, e.g., a drunkard, a frequent gambler, debt-loaded, etc.; (2) whether the plaintiffs were domestically abused by the defendants; (3) whether the defendants were domestically abused by the plaintiffs; (4) whether the plaintiffs took care of the family responsibly in the past; (5) whether the defendants were physically or mentally disabled; (6) whether the average monthly income of the defendants is above the national median monthly income; (7) whether the average monthly income of the defendants is below the minimum monthly salary (that is required by law). We summarize these factors in Table 4. The last two factors measure the relative economic conditions between the responsible persons and the defendants from both the perspective of the monthly incomes (RI) and their estates (RE).

5.2 Using A Model-Tree Model for Prediction

We show the main flow of our model tree in Figure 7. Note that, although the judges awarded plaintiffs in 938 cases among the 1930 cases that we found useful from TWJY, only 740 cases of them remain useful for the study of predicting the monthly support.

We define the **average request** in Equation (1). Based on the definitions of C, G, and O (in Table 3), we may consider that $C - G + O$ is the amount that the plaintiff needs to maintain an ordinary living. The ratio $\frac{P}{N}$ is the proportion that the plaintiff can request the defendants to support among the responsible persons.

$$\text{average request} = (\text{C} - \text{G} + \text{O}) \times \frac{P}{N} \quad (1)$$

Among the 740 cases that the judges would award grants to the plaintiffs, the requested amounts (A in Table 3) of 404 cases were larger than the average request. 336 cases were smaller. We then checked different subsets of the social factors (Table 4) for these two branches. For those 336 cases that had lower requests, 283 of them did not have any of the conditions from S1 to S5, so our model would predict that the judges would award the requested amount (A). For those 404 cases that had higher requests, 214 of them did not have any of the conditions from S1 to S7, so we used them to train and test a linear regression model, which we shall explain next.

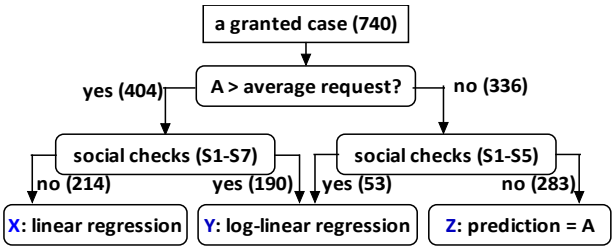


Fig. 7. The main flow of our model tree

In the lower middle of Figure 7, we can see that 53 cases that had lower requests and 190 cases that had higher requests would be used to train and test a log-linear regression model, because these cases satisfied at least one special social factor from S1 to S7. We shall explain the log-linear model below.

5.3 The Linear and Log-Linear Regression Models

We used the 214 cases (lower left corner of Figure 7) to train and test a linear regression model. We relied on the **LinearRegression** of the scikit learn for this task, and used the default settings. We employed the features listed in Table 3 in the experiments, used 80% of the 214 cases for training, and 20% for tests.

There are 243 cases (190+53, in the lower middle part of Figure 7) that we would use to train the log-linear regression model. The main purpose was to use the average request as the basis and to adjust the basis based on the factors that we listed in Table 4. Notice that factors S1 to S7 are Boolean and that RE and RI are ratios. When the Boolean factors are true, they will be converted to **2**. If they are false, they will be converted to **1**. Hence, if results at the step of "social checks" are [True, True, False, False, True, True, False, 1.2, 1.1], we will obtain a **feature vector** of [2, 2, 1, 1, 2, 2, 1, 1.2, 1.1].

Let us denote the feature vector that we obtain at the "social checks" as $F = \{f_1, f_2, \ldots, f_9\}$. We assume that the grants can be modeled by Equation (2), where A is defined in Table 3. The average request was defined in equation (1), and it would vary with each different case. Notice that when a Boolean feature is false, it is converted to 1. When an f_i is 1, the power of f_i will also be 1, independent of the value of w_i. Hence, when a social factor is false, that feature will **not** influence the calculated grant.

$$\text{grant} = \min(\text{A}, \text{average\_request}) \times \prod_{i=1}^{i=9} f_i^{w_i} \quad (2)$$

We can convert the exponential form in (2) by taking the logarithm of both sides.

Equation (2) will turn into a form of linear regression in Equation (3), and we could still rely on the LogisticRegression of scikit learn to train and test our models. Again, we split the data into the ratio of 8:2 for training and test, respectively.

Table 5. The averages MAE of 1000 experiments

test group	only linear regression	linear+log-linear	Fig. 7
X	-	1605.41	1048.71
Y	-	1968.96	1968.96
Z	-	3132.69	3112.20
All (740)	2390.27	2212.59	1992.88

$$\log(\text{grant}) = \log\big(\min(\text{A}, \text{average\_request})\big) + w_i \sum_{i=1}^{i=9} f_i \tag{3}$$

5.4 Results of Empirical Evaluations and Ablation Study

We used the mean absolute error (MAE) to measure the quality of the predictions for the grants. Let g_i and p_i denote the actual monthly grant and the predicted grant for a case i, respectively, the MAE is defined in (4), where n denote the number of test cases in an experiment. We reported the results of measuring the quality with other metrics in [10]

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{i=n}|g_i - p_i| \tag{4}$$

We repeated the experiments 1000 times, each by re-splitting the training and the test data. Table 5 shows the averages of the MAE values of these 1000 experiments. The unit of these numbers is Taiwanese dollars. The rightmost column shows the results of the average MAEs in the X, Y, and Z in Figure 7, which is the best performing model. If we did not include the "social checks (S1-S5)" step, and let the 336 cases be handled by the log-linear model in Figure 7, we would observe the results listed in the column "linear+log-linear", which are inferior to those that are listed in the column "Fig. 7". In this case, the X and the Y groups were used to train and test a log-linear regression model. We still could calculate and show their MAEs separately in Table 5. If we used all of the 740 cases as the training and test data for the linear regression model directly, we would observe the result listed under the column "only linear regression", which is the worst among the three designs.

We compared our results with those reported in [4], and found that the size of our dataset was larger, and our results were also better.

We can look more deeply into the results. Define the **degree of deviation** of the predicted grants from the actual grants in Equation (5). If a p_i is 10% larger or smaller than the true g_i for a case i, then the deviation will be 0.1.

$$\text{deviation}_i = \frac{|g_i - p_i|}{g_i} \tag{5}$$

Figure 8 shows the distribution of the deviation. The horizontal axis shows the ranges of the deviation, using 10% for the increments. The vertical axis shows the percentages of cases in the test data. In this small-data test, the predicted grants were within 10% range of the actual grants for more than 45% of the test cases.

6 Using NER for Factor Identification

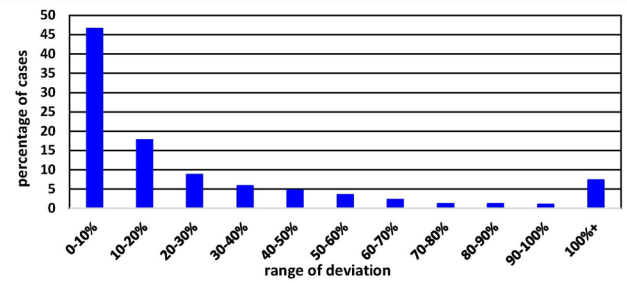


Fig. 8. The distribution of deviations

Table 6. The results of using NER for factor identification

Feature	C	G	O	A	I	E
F_1	0.915	0.840	0.676	0.821	0.868	0.960

The results discussed in the previous section relied on humans to pick the feature values from the judgment documents for the classifiers. We just showed that a prediction system may recommend grants that are fairly close to the judges' decisions, under such favorable precondition. Although this assumption seems impractical, it is how the online recommender system of the Judicial Yuan works.

The Judicial Yuan started to offer an online legal judgment prediction system for eight categories of criminal crimes many years ago. The system asks the human users to enter feature information about the criminal activities, and the system will provide a range and a recommended sentence for the provided information.[1] If we are holding a judgment documents and want to test this system, then our task would be to extract the correct information from the document, enter the information to the system, and see if the recommender will return good recommendations. That is what we did and described in the last section.

In this section, we report our attempt to push the boundary further. We try to identify the values of the key features algorithmically. We employed a tool for named-entity recognition (NER), W2NER [6].[5] With the name entities recognized by the W2NER, we can apply regular expressions to train and learn the functions of the numbers in the plaintiffs' and the defendants' claims. Following are a few sample contexts in which the functions of the numbers before "元" are about the social allowance (Table 3).

- 聲請人每月領有榮民就養給與(.{0,15})元
- 每月領有身心障礙者補助(.{0,15})元
- 目前領有托育養護補助每月(.{0,15})元
- 中(.{0,1})低收入戶(.{0,1})補助(.{0,15})元,

With the help of W2NER, we could achieve seemingly good results for the features that we listed in Table 3. Table 6 provides the F_1 measures.

Using the recognized feature values, we repeated the experiments that we reported in Section 5, and we observed the results in Table 7 and Figure 9. All of the predictions in Table 7 are much worse than their counterparts in Table 5. We have only about 35% of the cases for which the predicted values were within 10% of the correct answers.

[5] W2NER: https://github.com/ljynlp/W2NER, last accessed 2023/05/21

Recognizing some of the feature values correctly is not good enough for predicting the grant. When some of the feature values are wrong or missing, it is still hard to come up with correct or high-quality predictions with a fully automatic procedure.

Table 7. The averages MAE of 1000 experiments with NER

test group	only linear regression	linear+log-linear	Fig. 6
X	-	3080.29	2664.39
Y	-	3044.53	3044.53
Z	-	3533.94	7613.63
All (740)	3415.35	3171.19	3912.93

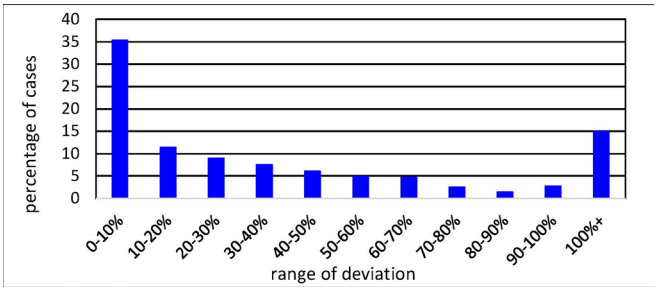


Fig. 9. The distribution of deviations with NER

7 Concluding Remarks

The reported observations in some of the experiments are encouraging, and some others show us more work to do. To provide more precise recommendations for judgments automatically, we need to find ways to identify the relevant factors more precisely from the text of the civil cases (and, of course, of the criminal cases). Gray et al. have just showed an example [3]. We should strengthen the depth of our current work as well [9].

Acknowledgments

We thank the reviewers for their recommendations about how we may improve our predictors. We will consider those additional factors in future experiments, although we could not do so for JURISIN 2023. This article was based on the thesis of Wei-Zhi Liu who was co-advised by Sieh-chuen Huang and Chao-Lin Liu [10]. Po-Hsien Wu assisted in the technical part, and Ho-Chien Huang offered comments in law. Wei-Zhi Liu's research also benefited from the comments of Chi-Yang Lin and Min-Yuh Day of National Taipei University, Taiwan. Chao-Lin Liu is the main author of this article. This research was supported in part by the grant 110-2221-E-004-008-MY3 of the National Science and Technology Council of Taiwan.

References

1. **Bex**, Floris and **Prakken**, Henry: On the relevance of algorithmic decision predictors for judicial decision making, *Proceedings of the 2021 International Conference on Artificial Intelligence in Law*, pp. 175–179, 2021.
2. **Feng**, Yi, **Li**, Chuanyi, and **Ng**, Vincent: Legal judgment prediction: A survey of the state of the art, *Proceedings of the 2021 International Joint Conference on Artificial Intelligence*, pp. 5461–5469, 2021
3. **Gray**, Morgan, **Savelka**, Jaromir, **Oliver**, Wesley, and **Ashley**, Kevin: Toward automatically identifying legally relevant factors, *Proceedings of the Thirty-Fifth International*

Conference on Legal Knowledge and Information Systems, pp. 53–62, 2022.

4. **Huang**, Sieh-chuen: An empirical study on the grants for the support for the elderly (老親扶養費酌定裁判之實證研究), presented in the Conference on Technology for Law and Justice (法律科技與接近正義研討會), National Taiwan University, 2022. (in Chinese)
5. **Komamizu**, Takahiro, **Fujioka**, Kazuya, **Ogawa**, Yasuhiro, and **Toyama**, Katsuhiko: Exploring relevant parts between legal documents using substructure matching, *Proceedings of the 2019 JURISIN*, pp. 5–19, 2019.
6. **Li**, Jingye, **Fei**, Hao, **Liu**, Jiang, **Wu**, Shengqiong, **Zhang**, Meishan, **Teng**, Chong, **Ji**, Donghong, and **Li**, Fei: Unified named entity recognition as word-word relation classification, *Proceedings of 2022 AAAI Conference on Artificial Intelligence*, pp. 10965–10973, 2022.
7. **Lin**, Chieh-Feng: *Regulation of the Judicial Discretion in Domestic Property Law: Focusing on the Determination of Maintenance, Living Expenses of the Household, and Alimony*, Dissertation of the College of Law, National Chengchi University, Taiwan, 2014. (in Chinese)
8. **Lin**, Kang-I: A *Computational Approach to Analyze Court Decisions Regarding Alimony Claims after Divorce*, Thesis of the Graduate Institute of Interdisciplinary Legal Studies, National Taiwan University, 2018. (in Chinese)
9. **Liu**, Chao-Lin, **Lin**, Hong-Ren, **Liu**, Wei-Zhi, and **Yang**, Chieh: Functional classification of statements of Chinese judgment documents of civil cases (alimony for the elderly), *Proceedings of the Thirty-Fifth International Conference on Legal Knowledge and Information Systems*, pp. 206–212, 2022.
10. **Liu**, Wei-Zhi: *Predicting Judgments and Grants for Civil Cases of Alimony for the Elderly*, Thesis of the Department of Computer Science, National Chengchi University, 2023. (in Chinese)
11. **Manning**, Christopher D., **Raghavan**, Prabhakar, and **Schütze**, Hinrich: *Introduction to Information Retrieval*, chapter 6, Cambridge University Press, 2008.
12. **Muhlenbach**, Fabrice, **Sayn**, Isabelle, **Nguyen-Phuoc**, Long: Predicting court decisions for alimony: avoiding extra-legal factors in decision made by judges and not understandable AI models, presented in ICML 2020 Workshop on Law and Machine Learning, 2020. arXiv:2007.04824, last accessed 2023/04/17
13. **Rabelo**, Juliano, **Kim**, Mi-Young, and **Goebel**, Randy: The application of text entailment techniques in COLIEE 2020, *Proceedings of the 2020 JURISIN*, pp. 240–253, 2020.
14. **Reimers**, Nils and **Gurevych**, Iryna: Sentence-BERT: Sentence embeddings using siamese BERT-networks, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the Ninth International Joint Conference on Natural Language Processing*, pp. 3982–3992, 2019.
15. **Wu**, Yiquan, **Kuang**, Kun, **Zhang**, Yating, **Liu**, Xiaozhong, **Sun**, Changlong, **Xiao**, Jun, **Zhuang**, Yueting, **Si**, Luo, and **Wu** Fei: De-Biased court's view generation with causality, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 763–780, 2020.
16. **Yamakoshi**, Takahiro, **Komamizu**, Takahiro, **Ogawa**, Yasuhiro, and **Toyama**, Katsuhiko: Differential translation for Japanese partially amended statutory sentences, *Proceedings of the 2020 JURISIN*, pp. 162–178, 2020.
17. **Zhao**, Lili, **Yue**, Linan, **An**, Yanqing, **Liu**, Ye, **Zhang**, Kai, **He**, Weidong, **Chen**, Yanmin, **Yuan,** Senchao, and **Liu**, Qi: Legal judgment prediction with multiple perspectives on civil cases, *Proceedings of 2021 CAAI International Conference on Artificial Intelligence*, pp. 712–723, 2021.

Encoding Defeasible Deontic Logic in Answer Set Programming

Guido Governatori and Meng Weng Wong

Centre for Computational Law, Singapore Management University, Singapore

Abstract. We present a brief overview of the Domain Specific Language L4, and we provide a defeasible semantics of it based on the Answer Set Programming encoding of Defeasible Deontic Logic.

1 Introduction

Defeasibilty is a concept whose importance for legal reasoning has been investigated for a long time [14, 4, 15]. The notion mostly concerns the issue that textual provisions of (legal) norms typically provide *prima facie* conditions for their applicability, but to understand a norm fully, we have to evaluate the norms in the context in which the norm is used and to see if other norms prevent it either to apply or to be effective. In other words, when evaluating norms, we must account for possible (*prima facie*) conflicts and exceptions. Indeed, in general, norms first provide the basic conditions for their applicability. Then, they give the exceptions and exclusions (and they can go on, with exceptions/exclusions of the exceptions/exclusions and so on).

The first issue to address to model legal reasoning is how to model norms. Here, we follow the approach of [17, 5] and stipulate that a norm is represented by an "IF $\cdots$ THEN $\ldots$" rule, where the IF part establishes the conditions of applicability of the norm. The THEN part specifies the legal effect of the norm. Where the legal effect of the norm is either that a proposition is taken to hold legally or that a legal requirement (obligation, prohibition, permission) is in force. Moreover, as we have alluded to, the norms are defeasible; thus, the IF/THEN conditional used to model legal norms does not correspond to the material implication of classical logic, and it has a non-monotonic nature. Several approaches have been proposed to reduce or compile the normative IF/THEN conditional. However, in general, as discussed by [18, 12], they suffer from some limitations; for example, the translation to classical propositional logic requires complete knowledge (for any atomic proposition we have to determine whether it is true or not), it is not resilient to contradictions, and changes to the norms might require a complete rewriting of the translation.

In this work, we are going to examine how to provide an effective and constructive non-monotonic interpretation of (a restricted version of) L4 based on an Answer Set Programming (ASP) meta-program. The meta-program gives the semantics and a computational framework for the underlying L4 constructs.

2 A Quick Overview of L4

L4 is a Domain Specific Language (DSL) proposed by the CCLAW project[1] to formalise law text. Current investigation of L4 has shown that the language is sufficiently precise to avoid ambiguities of natural languages and, at the same time, sufficiently close to a traditional law text with its characteristic elements such as cross-references, prioritisation of rules and defeasible reasoning. Moreover, once a law has been coded in L4, it can be further processed for different tasks for applications involving some form of legal reasoning.

This work focuses on one aspect of L4. More specifically, we concentrate on the basic notion of a rule and how to encode rules in an Answer Set based Defeasible Deontic Logic meta-program. A basic rule in L4 has the following form.

```
rule <r> if Preconditions then Conclusion
    {restrict: {subject to <s_1>,...,<s_n>}
               {despite <d_1>,...,<d_m>} }
    least Compensation
```

where `r`, a unique identified, is the label (name) or the rule; `Preconditions` is a (possibly empty) conjunction of propositions. Accordingly, we consider it as a set of propositions; `Conclusion` is a single proposition. The propositions in a rule can be prefixed by one of the following expressions: `MUST`, `MAY`, `SHANT` indicating the deontic modifier (operator) that applies to the proposition. The keyword `restrict` specifies what rules are either stronger (`subject to`) or weaker `despite` than the current rule. Finally, `Compensation` is a (deontic) proposition, and it represents the penalty or compensation for the violation of the norm encoded by the rule.

3 Defeasibility

Legal rules can be classified either as *constitutive rules* (also known as *counts as* rules) or *regulative rules*. In turn, a normative rule can either be a *prescriptive rule* or a *permissive rule*. A constitutive rule gives the meaning or defines a term; regulative rules specify what normative positions (obligations and prohibitions for prescriptive rules and permissions for permissive rule) and the conditions under which such normative positions hold. Defeasibility applies to both constitutive rules and regulative rules. Consider the following two real-life examples from Australian regulations and Acts.

Example 1 (Telecommunications Consumer Protections Code (C628:2012). Section 2.1. Definitions).

> **Complaint** means an expression of dissatisfaction made to a Supplier in relation to its Telecommunications Products or the complaints handling

[1] `https://cclaw.smu.edu.sg/`

process itself, where a response or Resolution is explicitly or implicitly expected by the Consumer.

An initial call to a provider to request a service or information or to request support is not necessarily a Complaint. An initial call to report a fault or service difficulty is not a Complaint. However, if a Customer advises that they want this initial call treated as a Complaint, the Supplier will also treat this initial call as a Complaint.

Example 2 (National Consumer Credit Protection Act 2009 (Act No. 134 of 2009). Section 29).

(1) A person must not engage in a credit activity if the person does not hold a licence authorising the person to engage in the credit activity.
(3) For the purposes of subsections (1) and (2), it is a defence if:
  (a) the person engages in the credit activity on behalf of another person (the principal); and
  (b) the person is:
    (i) an employee or director of the principal or of a related body corporate of the principal; or
    (ii) a credit representative of the principal; and ...

The semantics/computation for L4 rules we are going to present is based on Defeasible (Deontic) Logic [2, 11]. In Defeasible Logic, a proposition p holds (defeasibly) if:

- p is a fact; or
- there is a rule r such $Conc(r) = p$, and
  - for all $q \in Pre(r)$, q (defeasibly) holds (the rule is applicable), and
  - for any rule s such that $Conc(s) = \sim p$, s is either discarded or defeated

A rule s is *discarded* if there is proposition $q \in Pre(s)$ such that q is refuted, where refuted is the (constructive) failure to show that it holds. A rule s is *defeated* if there is an applicable rule t whose conclusion is the opposite of the conclusion of s and t is stronger than s.

The logic is sceptical in the sense that if there are two (applicable) rules for opposite conclusions and there are no means to solve the conflict, the logic prevents the conclusion of contradictions. Still, at the same time, it discards the conclusion of both conclusions. Hence, none of the two opposite conclusions holds. In other words, there is some ambiguity about which of the two conclusions hold. However, it is possible that the opposite conclusions can be part of the preconditions of other rules. Consider, for example, the scenario where there are two equally compelling different pieces of evidence, one supporting the case that a person was legally responsible for A and the second that the person was not responsible for A. Moreover, if the person was responsible for A, then the person is found guilty. However, according to the presumption of innocence, a person is assumed to be not guilty. This situation can be represented by

```
rule <r1> if evidence1 then responsible
rule <r2> if evidence2 then not responsible
rule <r3> if responsible then guilty
    {restrict: {despite <r4>}}
rule <r4> if true then not guilty
```

Given the two pieces of evidence, we cannot assert whether responsible or not responsible holds; thus, the proposition responsible is ambiguous. A statement is ambiguous when there is an argument supporting it and an argument for its opposite, and there are no means to determine if one of the two arguments is stronger/defeats the other. However, in this situation, we can go on since we cannot assert that responsible holds, but rule r_4 (encoding the so-called presumption of innocence) vacuously holds, and we can conclude `not guilty`.

However, Suppose that, in addition to the conditions stipulated above, if a person was wrongly accused, then the person is entitled to some compensation. This can be encoded in L4 by the following two rules:

```
rule <r5> if not guilty then compensation
rule <r6> if true then not compensation
    {restrict: {subject to <r5>}}
```

If we continue with our reasoning, rule r_5 is applicable; it defeats r_6, allowing us to establish that `compensation` holds. However, the two pieces of evidence were equally reliable; accordingly, it does not sound right that the person was wrongly accused. Indeed, it was ambiguous whether the accused was legally responsible or not. This scenario illustrates that we have to account for two forms of defeasibility: *ambiguity blocking* and *ambiguity propagation*. Governatori [6] argues that these two forms of defeasibility account for different (legal) proof standards. Defeasible Logic can accommodate the two variants.

The semantics we gave above is for the ambiguity blocking case. A few changes are needed for the ambiguity propagating case, [1]. First, a conclusion is *supported* if it is a fact or there is a rule such that all the preconditions are supported and the rule s not weaker than an applicable rule for the opposite. Second, rules attaching a conclusion are discarded if they are not supported (instead of applicable). These changes simplify attacking a conclusion, and it is easy to verify that both `guilt` and `not guilty` are supported, and we prevent the conclusion of `compensation`.

4 Defeasible Encoding of L4 in Answer Set Programming

In this section, we give a meta-program in Answer Set Programming to encode the reasoning mechanism we presented in the previous section to model defeasibility with both ambiguity blocking and ambiguity propagation. The ASP meta-program clauses that capture ambiguity blocking and ambiguity propagation are based on Defeasible Logic variants and meta-program given in [1]. The

deontic extension is based on the Defeasible Deontic Logic of [11] and the meta-program techniques of [13].[2] In addition to the meta-program clauses to model and compute the logic aspects, we discuss how to encode an L4 theory in the meta-program to compute the extension of the theory.

Propositions are declared with the `atom/1` predicate. The first step is establishing when two atomic propositions conflict, namely when two propositions cannot hold simultaneously. The ASP code we are going to present is a meta-program for the Defeasible Deontic Logic encoding of L4; accordingly, we cannot simply use the standard ASP negation (`-`) and negation as failure (`not`) to provide the negation of a term in L4/Defeasible Deontic Logic, but we have to introduce a new operator `non/1`. The next two clauses assert that the negation of `non(X)` is `X`, and the negation of `X` is `non(X)`.

```
negation(non(X),X) :- atom(X).
negation(X,non(X)) :- atom(X).
```

In addition to the negation, where we have that `p` and `non(p)` are the negation of each other, the language is equipped with the `conflict/2` predicates. For instance, `conflict(p,q)` allows us to assert that `p` cannot be true when `q` is. Notice that `conflict` is not symmetric; thus `conflit(p,q)` does not imply `conflict(q,p)`. The symmetric version is given by `strongConflict/2`.

```
conflict(X,Y) :- strongConflict(X,Y).
conflict(Y,X) :- strongConflict(X,Y).
```

Finally, `opposes/2` combines the two options to encode conflicting literals:

```
opposes(X,Y) :- negation(X,Y).
opposes(X,Y) :- conflict(X,Y).
```

The last element of the language is the superiority relation, a binary relation over rules that specify the relative strength of two rules. To this end, we have the predicate `superior/2`. The meaning of `superior(r,s)` is that rule `r` is stronger than rule `s`. Thus, if both rules are applicable, rule `r` defeats rule `s`. Also, we accommodate the other direction as well.

```
superior(X,Y) :- inferior(Y,X).
```

4.1 Modelling Ambiguity Blocking

We need the following clauses to model the computation for ambiguity blocking aspect of defeasibility. `fact(X)` indicates that `X` is a fact, an indisputable piece of evidence in a given case/situation. `defeasible(X)` means that the proposition `X` holds defeasibly.

[2] The source code for the encoding of Defeasible Deontic Logic in ASP is available at `https://github.com/gvdgdo/Defeasible-Deontic-Logic`.

```
defeasible(X) :-  fact(X).
defeasible(X) :-  opposes(X,X1), not fact(X1),
    rule(R,X), applicable(R), not overruled(R,X).
```

The predicate `applicable/1` takes as its argument a rule in L4, and its truth is determined by the ASP encoding of the L4 rule (we discuss the full procedure of how to encode an L4 rule below). Then, we can assert that an atom `X` holds defeasibly if it is asserted as a fact, or if there is an applicable rule for it, i.e., `rule(R,X)`, that is not overruled. In addition, there is no fact `X1` the atom `X` opposes to.

```
overruled(R,X) :- opposes(X,X1), rule(R,X),
    rule(R1,X1), applicable(R1), not defeated(R1,X1).
```

A rule `R` is overruled if there is a rule `R1` for a conclusion incompatible with the conclusion of `R`, and `R1`, in turn, is not defeated.

Finally, a rule `R` is defeated if there is a stronger applicable rule `R1` for the opposite.

```
defeated(R,X) :- opposes(X,X2), rule(R2,X2),
    superior(R2,R), applicable(R2).
```

4.2 Modelling Ambiguity Propagation

For ambiguity propagation, we need the following clauses (notice that the clauses below offer an alternative version of the `defeasible` predicate to the definition in Section ??).

Ambiguity propagation corresponds to ground semantics in argumentation [10]. To this end, we need to specify when a literal is *supported*, meaning that there is an undefeated argument for it. The idea is that we make it harder to establish that a rule is applicable and easier to attach a conclusion.

```
support(X) :- fact(X).
support(X) :- rule(R,X), supported(R), not beaten(R,X).
```

A proposition `X` is supported, `support(X)`, when the proposition is either a fact or there is a supported and not beaten rule for it. A rule is supported when all the elements in its antecedent are supported; similarly to what we have done for when a rule is applicable, this is formalised in the encoding of L4 rules.

```
beaten(R,X) :- rule(R,X), opposes(X,X1), fact(X1).
beaten(R,X) :- rule(R,X), opposes(X,X1), rule(R1,X1),
    supported(R1), superior(R1,R).
```

A rule `R` for `X` is `beaten` when the opposite of the conclusion is a fact or when there is a stronger supported rule for the opposite.

```
defeasible(X) :-  fact(X).
defeasible(X) :-  opposes(X,X1), not fact(X1),
    rule(R,X), applicable(R),  not overruled(R,X).
```

The conditions to establish that a proposition holds defeasibly are the same as those for ambiguity blocking (and so are the condition below for when a rule is defeated). The difference is when a rule is `overruled`.

```
overruled(R,X) :- opposes(X,X1), rule(R,X), rule(R1,X1),
    supported(R1), not defeated(R1,X1).

defeated(R,X) :- opposes(X,X2), rule(R2,X2),
    superior(R2,R), applicable(R2).
```

A rule `R` is defeated when there is a supported rule `R1` for the opposite of the conclusion of `R`.

Finally, the last two clauses allow us to state that every conclusion that holds defeasibly is supported and that every applicable rule is supported (see [3]).

```
support(X) :- defeasible(X).
supported(X) :- applicable(X).
```

4.3 Modelling Obligation and Permission

In this section we are going to provide the ASP encoding to model the Defeasible Deontic Logic proposed in [11]; to space reasons, we refer the readers to [11] for the description of the logic and its motivation. However, predicate names are self-describing, and the clauses provide an alternative description of the logic.

Rules are classified into three types of rules: *constitutive* rules encoding the definition of terms in a legal document; *prescriptive* rules for norms asserting their conclusion as an obligation (or prohibition), and *permissive* rules, producing a permission. In addition, prescriptive rules in Defeasible Deontic Logic have the following form

$$r\colon a_1, \ldots, a_n \Rightarrow_O c_1 \otimes \cdots \otimes c_m$$

where the meaning of the so-called compensation chain $c_1 \otimes c_2 \otimes \cdots \otimes c_m$ is that if the rule is applicable, c_1 is obligatory, but if it is violated, then the obligation of c_2 is in force and compensates the violation of c_1; similarly, if c_2 is violated, and c_3 compensates the violations of the previous obligation. We can repeat the same reasoning until c_m. Then, c_m is the last option to comply with the rule before we have a violation that cannot be compensated. The predicate `compensate(R,Y,X,N)` means that `Y` is an obligation in force, produced by rule `R`, after the obligation of `X` has been violated (and `X` appears at position `N` in the chain of compensations). Accordingly, for computation purposes, we will treat a compensation as a rule.

```
rule(R,X) :- constitutiveRule(R,X).
rule(R,X) :- prescriptiveRule(R,X).
rule(R,X) :- permissiveRule(R,X).
rule(R,X) :- compensate(R,_,X,_).
```

The next two clauses establish when a proposition can be asserted as a permission. The first is to capture the well-known obligation implies permission axiom of deontic logic. In contrast, the second captures the idea of weak permission when proving the opposite as an obligation is impossible.

```
permission(X) :- obligation(X).
permission(X) :- opposes(X,X1), not obligation(X1).
```

The ASP encoding features two `obligation` predicates, the first one, with a single argument `obligation/1`, specifying that a proposition holds as an obligation, and the second one `obligation/3`, with three arguments including the information about the rule used to derive the obligation and the position of the obligation in the compensation chain of the corresponding rule. The following clause gives the relation between the two predicates.

```
obligation(X) :- obligation(R,X,N).
```

The next two clauses encode two notions of violation.

```
violation(R,X,N):- obligation(R,X,N), opposes(Y,X), defeasible(Y).

terminalVioaltion(X) :- obligation(R,X,N),
    violation(R,X,N), not compensate(R,X,_,N).
```

A violation occurs when we have an obligation in force and the opposite of the content of the obligation holds defeasibly. Thus, for example, when for an atom `p` we have both `obligation(p)` and `defeasible(non(p))`. In addition, we have a terminal violation, when we have a violation, and the violated obligation is the last element of the reparation chain of the rule that entails the obligation.

The next block of clauses defines when a rule is applicable to entail the obligation of a proposition `X`. We have three cases: 1) there is a prescriptive rule `R` for `X` such that all the elements of the antecedent hold. 2) there is a constitutive rule for `X`, and the rule is "obligation applicable"; namely, the preconditions of the rule all hold as obligations. Finally, 3) there is an applicable rule `R` whose conclusion is a compensation chain and all the elements in the chain preceding `X` have been violated.

```
obligationApplicable(R,X,N) :- prescriptiveRule(R,X),
    applicable(R), N=1.
obligationApplicable(R,X,N) :- constitutiveRule(R,X),
    convertObligation(R), N=1.
obligationApplicable(R,X,N) :- compensate(R,Y,X,N-1),
    violation(R,Y,N-1).
```

The clauses for when we have a rule is able to produce a permission are similar to those for an obligation. However, permissive rules do not admit compensation chains, and thus we do not have the corresponding clause.

```
permissionApplicable(R,X) :- permissiveRule(R,X),
    applicable(R).
permissionApplicable(R,X) :- constitutiveRule(R,X),
    convertPermission(R).
```

As we presented before, the construction to assert a conclusion has an argumentation structure where in the first phase, we put forward an argument (applicable rules). Then we consider the attacking arguments, and finally, we rebut the attacking argument. An argument for an obligation or a permission corresponds to an obligation applicable rule or a permission applicable rule. The next step is to identify what are the attacking arguments/rules. An applicable obligation rule is attacked by either an obligation rule, a permissive rule, or a constitutive rule converting to obligation or permission, or compensation for the opposite conclusion.

```
obligationAttackingRule(R,X) :-
    prescriptiveRule(R,X), applicable(R).
obligationAttackingRule(R,X) :-
    permissiveRule(R,X), applicable(R).
obligationAttackingRule(R,X) :-
    constitutiveRule(R,X), convertObligation(R).
obligationAttackingRule(R,X) :-
    constitutiveRule(R,X), convertPermission(R).
obligationAttackingRule(R,X) :-
    compensate(R,Y,X,N), violation(R,Y,N).
```

In contrast, a permissive rule (or a rule behaving like a permissive rule) is attacked by an obligation rule, a

```
permissionAttackingRule(R,X) :-
    prescriptiveRule(R,X), applicable(R).
permissionAttackingRule(R,X) :-
    constitutiveRule(R,X), convertObligation(R).
permissionAttackingRule(R,X) :-
    compensate(R,Y,X,N), violation(R,Y,N).
```

Finally,

```
rebuttingRule(R,X) :- prescriptiveRule(R,X), applicable(R).
rebuttingRule(R,X) :- constitutiveRule(R,X), convertObligation(R).
rebuttingRule(R,X) :- compensate(R,Y,X,N), violation(R,Y,N).

obligation(R,X,N) :-
    obligationApplicable(R,X,N), not obligationOverruled(R,X).

obligationOverruled(R,X) :-
    opposes(X1,X), rule(R,X),
    obligationAttackingRule(R1,X1),  not obligationDefeated(R1,X1,X).
```

```
obligationDefeated(R,X1,X) :-
    opposes(X1,X), rule(R,X1), rebuttingRule(S,X), superior(S,R).

permission(X) :-
    permissionApplicable(R,X), not permissionOverruled(R,X).

permissionOverruled(R,X) :- opposes(X1,X), rule(R,X),
   permissionAttackingRule(R1,X1), not permissionDefeated(R1,X1,X).

permissionDefeated(R,X) :-
    opposes(X1,X), rule(R,X1), rebuttingRule(S,X), superior(S,R).
```

4.4 From L4 Rules to Their ASP Encoding

The process of encoding an L4 rule in the meta-program has the following step. First, we rewrite each `SHANT p` as `MUST not p`. Then we group all propositions in `Preconditions` in three groups (the groups can be empty): the first group is the set of propositions that do not occur in the scope of a deontic operator (`MUST`, `MAY`). The second and third groups are, respectively, the sets of propositions in the scope of '`MUST` and `MAY`. Thus a rule r has the generic form

```
rule <r>
    if a_1 && ... && a_n &&
       MUST o_1 && ... && MUST o_m &&
       MAY p_1 && ... && MAY p_k
    then [MUST|MAY] c
    {restrict: {subject to <s_1>...<s_l>} {despite <d_1>...<d_w>}}
    [least p]
```

A rule r is encoded as

```
prescriptiveRule(r,c). % if Conc(r) = MUST c
permissiveRule(r,c).   % if Conc(r) = MAY c
constitutiveRule(r,c). % otherwise

applicable(r) :-
    defeasible(a_1), ... , defeasible(a_n),
    % for each a_i not in the scope of MUST|MAY
    obligation(o_1), ... , obligation(o_m),
    % for each o_i in the scope of MUST
    permission(p_1), ... , permission(p_k).
    % for each p_i in the scope of MAY
```

In addition, for a constitutive rule r, i.e., rules where the conclusion is not in the scope of `MUST, MAY`, we have the clause

```
convertObligation(r) :- obligation(a_1), ..., obligation(a_n).
convertPermission(r) :- permission(a_1), ..., permission(a_n).
```

provided there is at least one `a_i`, there are no `o_i` in the scope of `MUST` and no `p_i` in the scope of `MAY`.[3]

For prescriptive rule, where there the `least` clause is not empty, we include

```
compensate(r,c,p,1).
```

Finally, we add

```
inferior(r,d). % for each d_i in restrict {despite <d_i>}
superior(s,r). % for each r_i in restrict {subject to <s_i>}
```

5 L4 and its Encoding at Work

Here we are going to illustrate how to use L4 and its ASP encoding with the help of an example taken from [8, 7]. The scenario has been recently used to test several rule-based implementations of legal reasoners [16].

Example 3.

Article 1. The Licensor grants the Licensee a licence to evaluate the Product.
Article 2. The Licensee must not publish the results of the evaluation of the Product without the approval of the Licensor. If the Licensee publishes results of the evaluation of the Product without approval from the Licensor, the material must be removed.
Article 3. The Licensee must not post comments on social media about the evaluation of the Product, unless the Licensee is permitted to publish the results of the evaluation.
Article 4. If the Licensee is commissioned to perform an independent evaluation of the Product, then the Licensee has the obligation to publish the evaluation results.
Article 5. This license terminates automatically if the Licensee breaches this Agreement.

The license can be represented in L4 by the following rules:

```
rule <r1> if True then SHANT use
rule <r2> if license then MAY use
    {restrict: {despite <r1>} {subject to <r7>}}
rule <r3> if True then SHANT publish
    least remove
rule <r4> if authorization then MAY publish
    {restrict: {despite <r3>}}
rule <r5> if True then SHANT social_media
rule <r6> if MAY publish then MAY social_media
```

[3] See [11] for the justification for such conditions.

```
    {restrict: {despite <r5>}}
rule <r7> if commission then MUST publish
rule <r8> if violation then SHANT use
    {restrict: {despite <r2>}}
```

Based on the procedure described in Section 4.4, the L4 rules are encoded in the following meta-program.

```
atom(use;license;publish;authorization;post;remove;commision).

prescriptiveRule(r1,non(use)).
applicable(r1).

permissiveRule(r2,use).
applicable(r2) :- defeasible(license).
superior(r2,r1).
inferior(r2,r8).

prescriptiveRule(r3,non(publish)).
applicable(r3).
compensate(r3,non(publish),remove,1).

permissiveRule(r4,publish).
applicable(r4) :- defeasible(authorization).
superior(r4,r3).

prescriptiveRule(r5,non(post)).
applicable(r5).

permissiveRule(r6,post).
applicable(r6) :- permission(publish).
superior(r6,r5).

prescriptiveRule(r7,publish).
applicable(r7) :- defeasiable(commission).

prescriptiveRule(r8,non(use)).
applicable(r8) :- terminalViolation(_).
superior(r8,r2).
```

Consider a case, where the Licensee publishes the results of the evaluation without the authorisation from the licensor, but remove it within the allotted time. This can be represented by `fact(license)`, `fact(publish)`' and `fact(remove)`. In this case, we obtain

```
obligation(remove) obligation(non(publish)) obligation(non(post))
```

However, if the licensor does not remove the published material in the given time, we reach a "terminal violation" indicating that there is a breach in the agreement, and the licence terminates automatically (i.e., it triggers the prohibition to use the product).

```
obligation(remove) obligation(non(use)) obligation(non(publish))
obligation(non(post)) terminalViolation(r3)
```

Notice that, in contrast with the ASP based Defeasible Deontic Logic implementation of L4, none of the implementations analysed in [16] is capable to deal with Article 5 (Automatic Termination). Moreover none of the other implementations offers a native treatment of the compensatory obligations, and have to depend on a contrary-to-duty representation (having to explicitly use obligation A and not A in the antecedent of a new rule). However, it is possible to give examples, of violation-based obligations that are not compensation [7].

6 Conclusion

As we discussed, ambiguity propagation and ambiguity blocking intuitions could be seen as different legal proof standards. A decision in a legal proceeding can use conclusions with different proof standards. The meta-programming approach presented in this paper allows us to accommodate them. For example, instead of using `defeasible/1`, we can replace it with `defeasible/2` where the first argument is the type of defeasible conclusion (ambiguity blocking or ambiguity propagation) and the second is the proposition. In addition, the encoding of a rule can specify what type of defeasibility is required for a precondition in given rules. We can have `defeasible(propagation,a_1), defeasible(blocking,a_2)` in one rule and `defeasible(blocking,a_1), defeasible(_,a_2)` in another rule. [9] proved that this combination is sound and complete and is a conservative extension of the individual variants. This shows that the defeasible deontic logic meta-programming encoding of L4 rules offers an efficient, flexible and powerful environment for modelling legal rules and a feasible and viable Rules as Code framework.

Acknowledgments

This research/project is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

1. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: A Flexible Framework for Defeasible Logics. In: AAAI 2000, pp. 401–405. AAAI/MIT Press, Menlo Park, CA (2000)

2. Antoniou, G., Billington, D., Governatori, G., Maher, M.J.: Representation Results for Defeasible Logic. ACM Trans. on Computational Logic 2(2), 255–287 (2001)
3. Billington, D., Antoniou, G., Governatori, G., Maher, M.J.: An Inclusion Theorem for Defeasible Logic. ACM Transactions in Computational Logic 12(1), article 6 (2010)
4. Bix, B.H.: Defeasibility and Open Texture. In: The Logic of Legal Requirements: Essays on Defeasibility, pp. 193–201. Oxford University Press (2012)
5. Gordon, T.F., Governatori, G., Rotolo, A.: Rules and Norms: Requirements for Rule Interchange Languages in the Legal Domain. In: Governatori, G., Hall, J., Paschke, A. (eds.) RuleML 2009, LNCS, vol. 5858, pp. 282–296. Springer, Heidelberg (2009)
6. Governatori, G.: On the Relationship between Carneades and Defeasible Logic. In: Ashley, K.D., van Engers, T.M. (eds.) The 13th International Conference on Artificial Intelligence and Law, pp. 31–40. ACM (2011)
7. Governatori, G.: Thou Shalt is not You Will. In: Atkinson, K. (ed.) Proceedings of the Fifteenth International Conference on Artificial Intelligence and Law, pp. 63–68. ACM, New York (2015)
8. Governatori, G., Idelberg, F., Milosevic, Z., Riveret, R., Sartor, G., Xu, X.: On legal contracts, imperative and declarative smartcontracts, and blockchain systems. Artificial Intelligence and Law 26(4), 377–409 (2018)
9. Governatori, G., Maher, M.J.: Annotated Defeasible Logic. Theory and Practice of Logic Programming 17(5–6), 819–836 (2017)
10. Governatori, G., Maher, M.J., Billington, D., Antoniou, G.: Argumentation Semantics for Defeasible Logics. Journal of Logic and Computation 14(5), 675–702 (2004)
11. Governatori, G., Olivieri, F., Rotolo, A., Scannapieco, S.: Computing Strong and Weak Permissions in Defeasible Logic. Journal of Philosophical Logic 42(6), 799–829 (2013)
12. Governatori, G., Padmanabhan, V., Rotolo, A., Sattar, A.: A Defeasible Logic for Modelling Policy-based Intentions and Motivational Attitudes. Logic Journal of the IGPL 17(3), 227–265 (2009)
13. Governatori, G., Rotolo, A.: Defeasible Logic: Agency, Intention and Obligation. In: Lomuscio, A., Nute, D. (eds.) DEON 2004, LNAI, vol. 3065, pp. 114–128. Springer, Heidelberg (2004)
14. Hart, H.L.A.: The Ascription of Responsibility and Rights. Proceedings of the Aristotelian Society 49, 171–194 (1948)
15. MacCormick, N.: Defeasibility in Law and Logic. In: Bankowski, Z., White, I., Hahn, U. (eds.) Informatics and the Foundations of Legal Reasoning, pp. 99–117. Springer Netherlands, Dordrecht (1995)
16. Robaldo, L., Batsakis, S., Calegari, R., Calimeri, F., Fujita, M., Governatori, G., Morelli, M.C., Pacenza, F., Pisano, G., Satoh, K., Tachmazidis, I., Zangari, J.: Compliance checking on first-order knowledge with conflicting and compensatory norms. A comparison among currently available technologies. Artificial Intelligence and Law (2023)
17. Sartor, G.: Legal Reasoning. Springer, Dordrecht (2005)
18. Sartor, G.: Normative Conflicts in Legal Reasoning. Artificial Intelligence and Law 1, 209–235 (1992)

Improving Vietnamese Legal Question–Answering System based on Automatic Data Enrichment

Thi-Hai-Yen Vuong[1], Ha-Thanh Nguyen[2], Quang-Huy Nguyen[1], Le-Minh Nguyen[3], and Xuan-Hieu Phan[1]

[1] VNU University of Engineering and Technology, Hanoi, Vietnam
{yenvth,19020011,hieupx}@vnu.edu.vn
[2] National Institute of Informatics, Tokyo, Japan nguyenhathanh@nii.ac.jp
[3] Japan Advanced Institute of Science and Technology, Ishikawa, Japan
nguyenml@jaist.ac.jp

Abstract. Question answering (QA) in law is a challenging problem because legal documents are much more complicated than normal texts in terms of terminology, structure, and temporal and logical relationships. It is even more difficult to perform legal QA for low-resource languages like Vietnamese where labeled data are rare and pre-trained language models are still limited. In this paper, we try to overcome these limitations by implementing a Vietnamese article-level retrieval-based legal QA system and introduce a novel method to improve the performance of language models by improving data quality through weak labeling. Our hypothesis is that in contexts where labeled data are limited, efficient data enrichment can help increase overall performance. Our experiments are designed to test multiple aspects, which demonstrate the effectiveness of the proposed technique.

Keywords: Vietnamese Legal QA, Data Enrichment, Legal Retrieval

1 Introduction

The performance of question-answering (QA) has increased significantly thanks to the rapid development and recent breakthroughs in natural language processing. With these advances, QA has been used actively in various business domains in order to save human labor, get more automation as well as enhance user experience. Among application areas, QA in the legal domain has attracted a lot of interest from the research community as well as the awareness and support from legal practitioners, experts, law firms, and government agencies. Legal QA could assist them to find relevant legal information quickly, accurately, and reliably.

Technically, the legal retrieval-based QA problem is simply stated as follows: given a query q and a text corpus $D = \{d_1, d_2, \ldots, d_n\}$, the retrieval-based QA finds the most likely document d^* that maximizes the relevance score R:

$$d^* = \arg\max_{d \in D} R(q, d) \tag{1}$$

where $R(q, d)$ represents the relevance score of the query q and document d.

Traditionally, lexical weighting and ranking approaches like TF-IDF or BM25 are used to find the relevant documents based on the match of vocabulary terms. Despite their limited accuracy and simplicity, these techniques are normally cost-effective. Meanwhile, representation and deep learning based models are likely to give better results but they are much more expensive in terms of large training data, computing power, storage, and deployment. Various deep learning models have been introduced to enhance the representation of queries and documents, such as CNN [4], RNN and LSTM [11,17]. Pre-trained language models (BERT [2], GPTs [1]) also significantly improve text representation in retrieval tasks.

In the legal domain, there are several challenges to building a reliable QA system. First, legal documents are much more complex than normal texts. They contain legal terms and concepts that are not commonly observed in general texts. Legal texts are usually long and have complex structures. There are also temporal constraints, logical relations, cross–document references etc. that are even difficult for human readers to follow and understand. Second, data annotation for legal documents is a real challenge, making it hard to construct even a medium-sized high-quality labeled dataset for training QA models.

Today, one popular way to improve accuracy is to build large deep-learning models with a huge number of parameters. This is obviously an obstacle because building such models requires powerful computing resources and a huge source of data. In this work, we want to concentrate on enhancing data quality and quantity in the context where expanding labeled data is infeasible. A heuristic method for automatically creating weak label datasets and supporting relationship representation models in case law retrieval is presented by Vuong et al. [20]. Therefore, we apply this technique to create more training data to improve our models without the need of increasing number of model parameters.

Technically, we address the problem of article-level retrieval-based legal QA. We use the Vietnamese civil law QA dataset, which was introduced by Nguyen et al. [10], to conduct an empirical study on the proposed methods. Table 1 illustrates an example of a legal query and the anticipated response. It is difficult to represent, retrieve and determine the correct answer when the articles are often long and complex. In addition, a notable feature of this dataset is that each article usually has a title, which serves as a brief summary.

The main contributions of our work are twofold. First, we built an end-to-end article retrieval system to solve the legal QA task. Second, we show how efficient automated data enrichment is and we conducted a variety of experiments to contrast our model with the most cutting-edge approaches in this domain.

2 Related Work

In natural language processing, the term question answering (QA) is commonly used to describe systems and models that are capable of providing information based on a given question. Depending on the characteristics of the task, we can divide it into different categories. Factoid QA [6] is a class of problems for which

Table 1: A sample in the dataset

Question	Hợp đồng ủy quyền có hiệu lực khi đáp ứng tiêu chí nào? *(An authorization contract is effective when it meets what criteria?)*
Answer	Article 117 form Document 91/2015/QH13
Article Title	Điều kiện có hiệu lực của giao dịch dân sự *(Valid conditions of civil transactions)*
Article Content	Giao dịch dân sự có hiệu lực khi có đủ các điều kiện sau đây: a) Chủ thể có năng lực pháp luật dân sự, năng lực hành vi dân sự phù hợp với giao dịch dân sự được xác lập; b) Chủ thể tham gia giao dịch dân sự hoàn toàn tự nguyện; c) Mục đích và nội dung của giao dịch dân sự không vi phạm điều cấm của luật, không trái đạo đức xã hội. Hình thức của giao dịch dân sự là điều kiện có hiệu lực của giao dịch dân sự trong trường hợp luật có quy định. *(A civil transaction takes effect when the following conditions are satisfied:* *a) The subject has civil legal capacity and civil act capacity suitable to the established civil transactions;* *b) Entities participating in civil transactions completely voluntarily;* *c) The purpose and content of the civil transaction do not violate the prohibition of the law and do not violate social ethics. The form of a civil transaction is the effective condition of a civil transaction in case it is provided for by law.)*

the answer is usually simple and can be further extracted from a given question or context. Problems in this category can often be solved with generation models or sequence tagging approaches. Retrieval-based QA [3] is a class of problems where the answer should be retrieved from a large list of candidates based on relevancy and ability to answer the question. This class of problems can also be called List QA. Confirmation QA [15] is the class of problems where systems or models need to confirm whether a statement is true or false. Systems for this type of problem can be an end-2-end deep learning model, knowledge-based systems, or neuro-symbolic systems.

In the legal field, question-answering has been posed in the research community for many years [12]. The main challenges of this problem on the rule language include fragmented training data, complex language, and long text. With the emergence of transformer-based [19] language models as well as transfer learning and data representation techniques, the performance of systems on tasks is significantly improved. In legal information retrieval, a number of neural approaches are also introduced to address the problem of word differences and characteristics of complex relationships [5,16,18,10].

3 Dataset

Original dataset: the corpus is collected from Vietnamese civil law. The labeled dataset was introduced by Nguyen et al. [10]. Table 2 & 3 give a statistical

summary of the corpus and dataset. There are 8587 documents in the corpus. Vietnamese civil law documents have a long and intricate structure. The longest document contains up to 689 articles, and the average number of articles per document is also comparatively high at 13.69. The average title length in this dataset is 13.28 words, whereas the average content length is 281.83 words.

Table 2: Corpus of Vietnamese legal documents statistics

Attribute	**Value**
Number of legal documents	8,587
Number of legal articles	117,557
Number of articles missing title	1,895
The average number of articles per document	13.69
Maximum number of articles per document	689
The average length of article title	13.28
The average length of article content	281.83

This is also worth noting because one of the challenges and restrictions is the presentation of long texts. On average, the questions are less than 40 words long. Because of the similarity in their distributions, it is expected that the model trained on the training set will yield good performance on the test set.

Table 3: Original dataset statistics

	Train set	**Test set**
Number of samples	5329	593
Minimum length of question	4	5
Maximum length of question	45	43
Average length of question	17.33	17.10
Minimum number of articles per query	1	1
Maximum number of articles per query	11	9
Average number of articles per query	1.58	1.60

Weak labeled dataset: Vuong et al. have the assumption that the sentences in a legal article will support a topic sentence [20]. On the basis of this supposition, the weak labeled dataset is created. There is also a similar relationship in this dataset. The title serves as a brief summary of the article, so the sentences in the article content support to title. We apply this assumption to our method. By considering the title to be the same as the question, we will produce a dataset with weak labels. A title and content pair would be a positive example equivalent to a question and related articles pair. We randomly generated negative examples at a ratio of 1:4 to positive labels and obtained a weak label dataset consisting of 551,225 examples.

4 Methods

For a legal question-answering system at the article level, given a question q, and a corpus of Civil Law $CL = \{D_1, D_2, ..., D_n\}$, the system should return a list of related articles $A = \{a_i | a_i \in D_j, D_j \in CL\}$. The following section provides a detailed description of the phases involved in resolving the problem.

4.1 General Architecture

Figure 1 demonstrates our proposed system. There are three main phases: preprocessing, training, and inference phase.

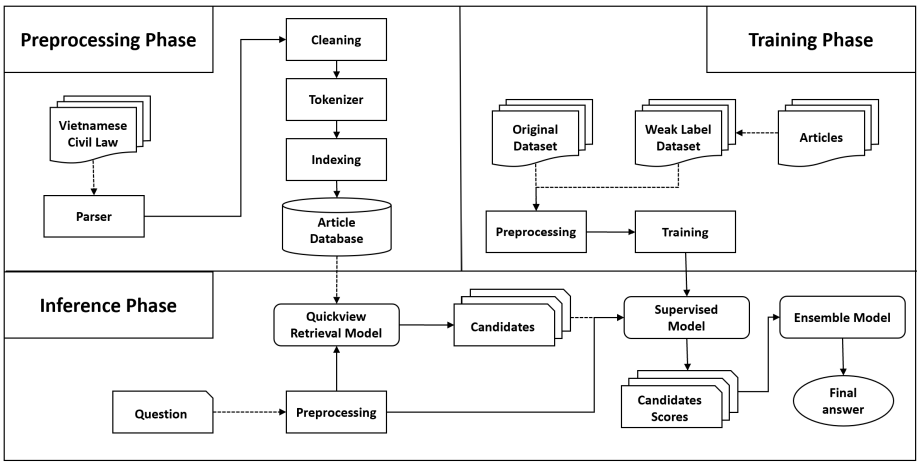


Fig. 1: Pipeline in the end-to-end article retrieval-base question answering system

Preprocessing phase: the result of this phase is an article-level database, which involves processing the raw Vietnamese civil law documents.

- ***Vietnamese Civil law*** is a corpus of Vietnamese legal documents.
- ***Parser*** segment legal documents into list of articles.
- ***Cleaning*** will filter out documents with metadata. Special symbol characters are also removed from the article. Numbers and vocabulary are retained and converted to lowercase.
- ***Tokenizer*** is crucial to the processing of Vietnamese natural language. Vietnamese word structure is quite complicated, a word might contain one or more tokens.
- ***Indexing*** is a task to represent and put articles into the database. Given a query, the search engine will return the response quickly and accurately.

Training phase: we construct a supervised machine-learning model to rank the articles pertaining to the input question.

- ***Original dataset*** is a legal QA dataset provided by Nguyen et al. [10].
- ***Articles*** is result of the preprocessing phase.
- ***Weak label dataset*** was create by our heuristic method.
- ***Preprocessing*** includes tasks similar to the preprocessing phase for question processing.
- ***Training***, we will construct a deep learning model to rank the texts related to the question.

Inference phase: is the process to generate the response to a new input question.

- ***Question*** is query in natural language.
- ***Preprocessing*** is same as previous phases to process input question.
- ***Quickview retrieval model*** matches questions and texts using unsupervised machine learning techniques . The processing speed of this model is typically fast.
- ***Candidates*** are a list of limited candidates returned from quickview retrieval model.
- ***Supervised model*** is result of the training phase. Its inputs are the question and the article candidates.
- ***Candicate scores*** are outputs of Supervised model.
- ***Ensemble model*** will combine the scores of the quickview retrieval model and the supervised model to make a final decision.

4.2 Indexing

There are numerous methods for indexing text into a database; in this work, we conduct experiments in two ways: word indexing and dense indexing.

Word indexing: During the indexing process, the words in the text will be analyzed, normalized, and assigned a corresponding index. When given a query, the system searches the index the most related. Word indexing helps to find and look up information in the text faster and more accurately.

Dense vector indexing: In addition to word indexing, word-to-vec and sequence-to-vec are both common methods for representing text semantically. These dense vectors can be used to represent text and index the database for search purposes. We apply two ways of representing text as dense vector according to w2v (FastText [7]) and contextual embedding (BERT [2]) to encode the given question and the legal articles. FastText is a model that converts each word into a dense vector of 300 dimensions. To construct a vector representation of a text, we average over the word vectors to form a single representation vector. Sentence-BERT converte the text into a dense vector with 768 dimensions that can represent the contextual semantics of the document by the Sentence-BERT model [13].

Table 2 shows that the length of articels is often large, which is a limitation of the text representation by FastText and BERT. On the other hand, most questions just partially match articles, we overcome this long presentation weakness by splitting the legal article into a list of sentences and then generating dense vectors before indexing them into the database.

4.3 Quickview Retrieval Model

There are 117,575 legal articles in this corpus. This is a huge number, so in order to ensure the effectiveness of the question-answering system, we build a so-called quickview retrieval model using unsupervised machine learning techniques in order to rapidly return a limited candidate set.

Word matching: to compare questions and articles in the word indexing database, we use the BM25 algorithm [14]. The bag-of-words retrieval function BM25 estimates the relevance of a document to a given search query by ranking documents according to the query terms that appear in each document.

Given a question Q, containing tokens $\{t_1, t_2, ..., t_n\}$, the BM25 score of a article A is:

$$BM25S(Q, A) = \sum_{i=1}^{n} IDF(t_i) \cdot \frac{f(t_i, A) \cdot (k_1 + 1)}{f(t_i, A) + k_1 \cdot (1 - b + b \cdot \frac{|A|}{avgdl})} \tag{2}$$

in which:

- $f(t_i, A)$: t_i's term frequency in the legal article A
- $|A|$: a number of word in in the legal article A in terms
- $avgdl$: the average article length in the legal corpus.
- k_1: a saturation curve parameter of term frequency.
- b: the importance of document length.
- $IDF(t_i)$ is the inverse document frequency weight of the given question t_i, follow as: $IDF(t_i) = \ln(1 + \frac{N - n(t_i + 0.5)}{n(t_i) + 0.5})$. N is amount of articles in the legal corpus, and $n(q_i)$ is amount of articles containing q_i.

While a content article is intense with full meaning, the article title contains a significant meaning. In this instance, the quickview retrieval score is determined using the formula below:

$$QS(Q, A) = \alpha * BM25S(Q, TA) + \beta * BM25S(Q, CA) \tag{3}$$

in which, α and β are boosting weights. TA and CA are the titles and content of the article.

Dense vector matching: to estimate the semantic similarity between questions and legal articles in the dense indexing database, we use cosine similarity to calculate quickview retrieval score:

$$Cosine(VQ, VSA) = \frac{VQ^T \cdot VSA}{||VQ|| \cdot ||VSA||} \tag{4}$$

$$QS(Q, A) = \max_{1 \le j \le n} (Cosine(VQ, VSA_j)) \tag{5}$$

in which, VQ is presentation vectors of the question.0. VSA_j is presentation vectors of the j^{th} sentence in the legal article. n is the number of sentences in the legal article. Finally, We use minmaxscaler to normalize scores and generate a list of ranked candidates.

4.4 Supervised Model

Pre-trained language models have proven useful for natural language processing tasks. Particularly, BERT significantly enhanced common language representation [2]. We use the BERT pre-training model and adjust all its parameters to build the related classifier model. We use the first token's final hidden state h as the presentation for the question-article pair. The last layer is a single fully connected added on the top of BERT. The output of the model is a binary classification. Cross-entropy loss is applied to the loss function. Adam [9] is used to optimize all model parameters during the training phase with a learning rate of e^{-5}. The supervised score between the question and the legal article is the classification probability of label 1:

$$SS(Q, A) = P_{label=1}(Q, A) \tag{6}$$

Lastly, we also use minmaxscaler to normalize scores and reranking a list of candidates. In this model, we proceed to build a related classification model based on two training datasets: the original dataset and a full dataset (original and weak label dataset). In the training process with the full dataset, we fit the model on weak label data first. Then use the best model to fine-tune with the original dataset.

4.5 Ensemble Model

We utilize the quickview retrieval model to generate a list of the $top - k$ candidates. These candidates are then refined using a supervised ensemble model, which provides higher precision but is slower. The quickview model serves as a preliminary selection step due to its fast computation despite its lower precision.

We use a variety of measures of similarity, including lexical similarity (the quickview retrieval model) and semantic similarity (the supervised model). Despite the fact that lexical and semantic similarities are very different from one another, they can work in tandem and are complementary. The combined score of the question Q and the candidate article CA_i is calculated as follows:

$$CombineS(Q, CA_i) = \gamma * QS(Q, CA_i) + (1 - \gamma) * SS(Q, CA_i) \tag{7}$$

where $\gamma \in [0, 1]$.

Table 2 indicates that each question can have one or more related articles (the average is about 1.6). The most relevant article $MRCA$ is returned by default, to determine a set of candidates to return, we would normalize the combined score and use the threshold parameter: a final returned articles set $FRA = \{CA_i | CombineS(Q, MRCA) - CombineS(Q, CA_i) < threshold\}$.

5 Experimental Results and Discussion

To ensure fairness in the training process and selection of hyperparameters, we divided the training dataset into training and validation with a ratio of 9:1. In

the quickview retrieval phase, we utilize the *Recall@k* measure to assess the list of returned candidates. Recall@k is (Number of correctly predicted articles in the $top - k$ results) / (Total number of gold articles). Macro-F2 is a metric to evaluate the end-to-end question-answering system. Precision, recall, and average response time per question are also used to evaluate the system's performance.

The processing phase and the quickview retrieval model are carried out on CPU Intel core i5 10500 and 32Gb ram. The supervised model is trained and inference on NVIDIA Tesla P100 GPU 15Gb. In the indexing step and the quickview retrieval model, we use Elasticsearch[4] with the configuration setting 8Gb heap size. Besides, during the experiment with some pre-trained BERT models, the BERT multilingual model produces the best results, so it is used to generate vector representation for the given question and the articles in the dense vector indexing and is used in a supervised model.

5.1 Quickview Retrieval Result

Table 4 shows the results of the word matching method, it is easy to see the superiority in execution time. It only takes 14.43 ms to return the set of 50 candidates and 115.63 ms for 1000 candidates. The results also demonstrate how the title and content of the article have an impact on the retrieval. *Recall*@1000 is only from 0.75 to 0.87 on the datasets if we solely utilize word matching based on either title or content. While using both of them, *Recall*@1000 is nearly 0.9. As a sort of written summary, the title frequently includes important keywords. Consequently, we achieve the best results, 0.9128 in *Recall*@1000, when increasing the question-title matching score by 1.5 times compared to the question content.

Table 4: *Recall@k* of Word matching method in quickview retrieval model

BM25(α, β)/Top-k	**50**	**100**	**200**	**500**	**1000**
Time per Q (ms)	14.43	20,32	31.32	63.21	115,63
Training set					
BM25(0,1)	0.6169	0.6941	0.7586	0.8220	0.8659
BM25(1,0)	0.5674	0.6169	0.6644	0.7172	0.7536
BM25(1,1)	0.6739	0.7478	0.8060	0.8651	0.8998
BM25(1.5,1)	0.6942	0.7612	0.8169	0.8740	0.9063
Testing set					
BM25(0,1)	0.6309	0.7103	0.7709	0.8368	0.8747
BM25(1,0)	0.5743	0.6282	0.6792	0.7259	0.7611
BM25(1,1)	0.6943	0.7728	0.8261	0.8798	0.9080
BM25(1.5,1)	**0.7214**	**0.7973**	**0.8453**	**0.8863**	**0.9128**

The experimental result of the dense vector matching method is illustrated in Table 5. Both the dense vector matching on BERT and FastText have lengthy

[4] https://www.elastic.co/

execution times but just average *Recall@k*. In the dense vector indexing method, the articles were indexed at the sentence level, we need to return larger records than the word indexing method based on article level. Calculating the similarity between vectors with large dimensions is also a challenge. Therefore, this method takes a long time to execute. Retrieving 10000 sentences that take 1.7 and 5,2 seconds is not possibly applied in the real-time question-answering system. *R*@10000 is 0.61 for the FastText and 0.67 for the BERT, It is also simple to understand these scores. because the advantage of FastText is a semantic representation at the word level. Whereas BERT is known for its powerful contextual representation of paragraphs, splitting the article into sentences loses this contextual property.

Table 5: *Recall@k* of quickview retrieval model on the dense vector indexing

k	EmbeddingMethod	R@k	Time(ms)
1000	FastText(D=300)	0.40	203
	BERT(D=768)	0.38	755
2000	FastText(D=300)	0.48	384
	BERT(D=768)	0.45	1,059
5000	FastText(D=300)	0.56	896
	BERT(D=768)	0.60	2,433
10000	FastText(D=300)	0.61	1,757
	BERT(D=768)	0.67	5,204

Based on the aforementioned experiment results, we decided to build the quickview retrieval model using BM25 with the $\alpha = 1.5$ and $\beta = 1$. For the real-time response, we obtain respectable *Recall@k* scores of 0.7214, 0.7973 and 0.8453 for the k values in (50, 100, 200), which indicates that the number of candidates will be returned following this phase.

5.2 End-to-end Question Answering System Result

Table 6 indicates the experimental results of the end-to-end question answering system result with a top 200 candidates from the quickview retrieval model. The word-matching model with BM25 and the supervised model built from the original data gives F2 score is about 0.38. The ensemble model outperforms the other models in F2 score with 0.6007, which is 22% higher than the single models. As was pointed out in the previous section, lexical and semantic similarity are highly dissimilar. But we believe they can cooperate and support one another. Results certainly support that. Table 6 also clearly illustrates the contribution of the weak label dataset. It improved the supervised machine learning model's F2 score by 8%. The weak label data continues to have an impact on the F2 score when the lexical and semantic matching models are combined. The ensemble model that used the weak label data had a 1% increase in F2 scores.

Table 6: The result of end-to-end QA system result with $top_k = 200$

Model	R	P	F2
Quickview Model(1.5,1)	0.4454	0.2399	0.3803
Supervised Model (original data)	0.6165	0.1461	0.3750
Supervised Model (full data)	0.6651	0.1998	0.4538
Ensemble Model (original data)	**0.6681**	0.4080	0.5925
Ensemble Model (full data)	0.6651	**0.4331**	**0.6007**

Additionally, there is a sizeable distinction between precision and recall. The recall is given more consideration because of its great impact on F2 score. We discovered that similarity in lexical and semantics has the same effect during the experimental and evaluation phases. Consequently, γ is set at 0.5. Infer time is also a remarkable point in the construction of the question-answering system, which shows the feasibility of the system when applied in practice.

Table 7 illustrate the results with the computational resources in the experimental environment, we can use the model with the top 50—100 candidates with an execution time of 1 second and 1.7 seconds per question. Their F2 scores are also only 2-5% lower than the best model.

Table 7: The result of end-to-end QA system result with ensemble model

Ensemble Model	R	P	F2	Time(s)
(full data, k=20)	0.5677	0.4034	0.5252	0.5
(full data, k=50)	0.5842	0.4428	0.5491	1
(full data, k=100)	0.6222	**0.4475**	0.5771	1.7
(full data, k=200)	0.6651	0.4331	**0.6007**	3.4
(full data, k=500)	**0.6793**	0.4015	0.5967	8.5
(full data, k=1000)	0.6583	0.4261	0.5936	17

Table 8 shows that our recall and F2 scores are incredibly high when compared to the Attentive CNN [8] and the Paraformer [10] models (0.6651 and 0.6007). Their models return small amounts of related articles, while our system is designed to return flexible amounts of articles with *threshold*. This explains why their precision is great, about 0.5987, whereas our precision is only 0.4331. A set of thresholds for each $top - k$ is listed in Table 9.

Table 10 describes an example of our legal question-answering system, compared with Paraformer [10]. A small number of related articles are frequently returned by Paraformer models. Our system is more flexible with 3 returned related articles. While the gold label number is 2. As an outcome, a paragraph model like Paraformer is produced that has great precision but low recall, whereas our method leans in the opposite direction. Since recall has a greater impact on F2 scores, our model has a significantly higher F2 score of 11%.

Table 8: The result compared with other research groups

Systems	**R**	**P**	**F2**
Attentive CNN [8]	0.4660	0.5919	0.4774
Paraformer [10]	0.4769	**0.5987**	0.4882
Our model (k=50)	0.5842	0.4428	0.5491
Our model (k=100)	0.6222	0.4475	0.5771
Our model (k=200)	**0.6651**	0.4331	**0.6007**

Table 9: Threshold list of the ensemble model

top_k	20	50	100	200	500	1000
threshold	0.38	0.28	0.26	0.26	0.25	0.2

Table 10: An output example of ours System, compared with Paraformer [10].

Question: Vay tiền để kinh doanh nhưng không còn khả năng chi trả phải trả lãi suất thì như thế nào? *(In the case of insolvency, how does one address the issue of paying the interest on a business loan?)*	Ours	Para-former	Gold
Candidate 1: Id: Article 357 from Doc 91/2015/QH13 **Title:** Trách nhiệm do chậm thực hiện nghĩa vụ trả tiền *(Liability for late performance of the obligation to pay)* **Content:** 1. Trường hợp bên có nghĩa vụ chậm trả tiền thì bên đó phải trả lãi đối với số tiền chậm trả tương ứng với thời gian chậm trả. 2. Lãi suất phát sinh do chậm trả tiền được xác định theo thỏa thuận của các bên nhưng không được vượt quá mức lãi suất được quy định tại khoản 1 Điều 468; nếu không có thỏa thuận thì thực hiện theo quy định tại khoản 2 Điều 468. *(1. Where the obligor makes late payment, then it must pay interest on the unpaid amount corresponding to the late period.* *2. Interest arising from late payments shall be determined by agreement of the parties, but may not exceed the interest rate specified in paragraph 1 of Article 468 of this Code; if there no agreement mentioned above, the Clause 2 of Article 468 of this Code shall apply.)*	1	1	1
Candidate 2: Id: Article 466 from Doc 91/2015/QH13 **Title:** Nghĩa vụ trả nợ của bên vay *(Obligations of borrowers to repay loans)* **Content:** [...]5. Trường hợp vay có lãi mà khi đến hạn bên vay không trả hoặc trả không đầy đủ thì bên vay phải trả lãi như sau: a) Lãi trên nợ gốc theo lãi suất thỏa thuận trong hợp đồng tương ứng với thời hạn vay mà đến hạn chưa trả; trường hợp chậm trả thì còn phải trả lãi theo mức lãi suất quy định tại khoản 2 Điều 468 của Bộ luật này; b) Lãi trên nợ gốc quá hạn chưa trả bằng 150% lãi suất vay theo hợp đồng tương ứng với thời gian chậm trả, trừ trường hợp có thỏa thuận khác. *([...] 5. If a borrower fails to repay, in whole or in part, a loan with interest, the borrower must pay:* *a) Interest on the principal as agreed in proportion to the overdue loan term and interest at the rate prescribed in Clause 2 Article 468 in case of late payment;* *b) Overdue interest on the principal equals one hundred and fifty (150) per cent of the interest rate in proportion to the late payment period, unless otherwise agreed.)*	1	0	0
Candidate 3: Id: Article 468 from Doc 91/2015/QH13 **Title:** Lãi suất *(Interest rates)* **Content:** 1. Lãi suất vay do các bên thỏa thuận.[...] 2. Trường hợp các bên có thỏa thuận về việc trả lãi, nhưng không xác định rõ lãi suất và có tranh chấp về lãi suất thì lãi suất được xác định bằng 50% mức lãi suất giới hạn quy định tại khoản 1 Điều này tại thời điểm trả nợ. *(1. The rate of interest for a loan shall be as agreed by the parties.[...]* *2. Where parties agree that interest will be payable but fail to specify the interest rate, or where there is a dispute as to the interest rate, the interest rate for the duration of the loan shall equal 50% of the maximum interest prescribed in Clause 1 of this Article at the repayment time.)*	1	0	1

Our model predicts that "Article 466 from Doc 91/2015/QH13" is relevant to the given query but the gold label is 0. Considering this article, we believe the article is pertinent to the given question but it seems that the annotator's point of view is different. In addition, we discovered some similar cases in our error analysis. Defining and agreeing on a measure of relevance is an important research question that needs the participation of the AI and Law community in its research. This not only benefits the development of automated methods but also makes legal judgments and decisions more reliable and accurate.

6 Conclusions

In this paper, we present a method to improve performance in the task of legal question answering for Vietnamese using language models through weak labeling. By demonstrating the effectiveness of this method through experiments, we verify the hypothesis that improving the quality and quantity of datasets is the right approach for this problem, especially in low-resource languages like Vietnamese. The results of our work can provide valuable insights and serve as a reference for future attempts to tackle similar challenges in low-resource legal question-answering.

Acknowledgement

This work was supported by VNU University of Engineering and Technology under project number CN22.09.

References

1. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL. pp. 4171–4186 (Jun 2019)
3. Feldman, Y., El-Yaniv, R.: Multi-hop paragraph retrieval for open-domain question answering. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2296–2309 (2019)
4. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Advances in neural information processing systems. pp. 2042–2050 (2014)
5. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. pp. 2333–2338 (2013)
6. Iyyer, M., Boyd-Graber, J., Claudino, L., Socher, R., Daumé III, H.: A neural network for factoid question answering over paragraphs. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 633–644 (2014)

7. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 427–431. Association for Computational Linguistics (April 2017)
8. Kien, P.M., Nguyen, H.T., Bach, N.X., Tran, V., Le Nguyen, M., Phuong, T.M.: Answering legal questions by learning neural attentive text representation. In: Proceedings of the 28th International Conference on Computational Linguistics. pp. 988–998 (2020)
9. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (12 2014)
10. Nguyen, H.T., Phi, M.K., Ngo, X.B., Tran, V., Nguyen, L.M., Tu, M.P.: Attentive deep neural networks for legal document retrieval. Artificial Intelligence and Law pp. 1–30 (2022)
11. Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., Ward, R.: Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. IEEE/ACM Transactions on Audio, Speech, and Language Processing **24**(4), 694–707 (2016)
12. Rabelo, J., Goebel, R., Kim, M.Y., Kano, Y., Yoshioka, M., Satoh, K.: Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. The Review of Socionetwork Strategies **16**(1), 111–133 (2022)
13. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992 (2019)
14. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: SIGIR'94. pp. 232–241. Springer (1994)
15. Sanagavarapu, K., Singaraju, J., Kakileti, A., Kaza, A., Mathews, A., Li, H., Brito, N., Blanco, E.: Disentangling indirect answers to yes-no questions in real conversations. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4677–4695 (2022)
16. Sugathadasa, K., Ayesha, B., de Silva, N., Perera, A.S., Jayawardana, V., Lakmal, D., Perera, M.: Legal document retrieval using document vector embeddings and deep learning. In: Intelligent Computing: Proceedings of the 2018 Computing Conference, Volume 2. pp. 160–175. Springer (2019)
17. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 1556–1566. Association for Computational Linguistics, Beijing, China (Jul 2015)
18. Tran, V., Le Nguyen, M., Tojo, S., Satoh, K.: Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. Artificial Intelligence and Law **28**, 441–467 (2020)
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
20. 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. Artificial Intelligence and Law pp. 1–28 (2022)

Legal Judgment Clustering Using Acts

Vishnuprabha V[1], Daleesha M Viswanathan[1], and Rajesh R[2]

[1] Cochin University of Science and Technology, Kerala, India
vishnuprabha72@gmail.com
[2] Naval Physical Oceanographic Laboratory

Abstract. Finding similar court cases is an important research problem in the legal domain. Developing automatic approaches for computing similar legal documents or clustering legal documents can greatly help the legal community. There were efforts to cluster the legal documents using citations, keywords, topics, catchphrases, etc. This work proposes a new approach to cluster the legal documents by exploiting the statutes/Acts in the legal judgment. The approach is evaluated manually using the Acts field for finding cluster purity. The results show that the statute information in the legal judgment is one of the best to cluster legal documents.

Keywords: Legal Judgments · Clustering · Legal Acts.

1 Introduction

The Indian court system follows a hybrid legal system involving civil and common law. Civil law uses established rules such as statutes, and common law uses precedents to solve a legal dispute. In a country like India that follows a hybrid law system, the lawyers use previous judgments with a similar case scenario along with statutes. Nevertheless, finding such a prior judgment that is similar to the existing scenario is a challenging task. However, as the judgments in India do not contain any title information, it is not easy to find similar judgments. To find the similarity, one must go through the headnotes or the entire judgment or paragraphs or other methods [9] to find the similarity.

Finding similar cases is one of the ultimate research aims in the legal industry. Computer experts think clustering or classification is the most efficient way to deal with large amounts of data [10]. Clustering structured information can further help in information retrieval and other tasks. Information retrieval is a popular research area in the Indian legal system. The characteristics of legal information always make the information retrieval task difficult [11]. Not just the characteristics, the availability of a large amount of unstructured information influence the retrieval task.

According to [10], Cluster Analysis involves four steps. Feature selection or extraction, Clustering algorithm design or selection, Cluster validation, Results interpretation. There were attempts [1, 21] to cluster the Indian legal judgments using citations and paragraph links. The authors [9]exploited information such

as whole document, RFC, headnote, and paragraphs and citations for finding similarity. They found that the whole document is the best approach to finding the similarity as getting other information, such as RFC, headnotes, etc., is expensive.

This paper focuses on the Indian legal judgment clustering where labelled data is unavailable. Another aim was to find the best feature for domain-specific clustering in the legal domain. The information such as Acts, Acts with their section information, Citations, and Whole judgment is considered for clustering. Here, the proposed approach is applied to a pool of 2020 Indian Supreme Court Case judgments scraped from the Legal Information Institute of India (LIIofIndia) website. The study found that Act is one of the best features for domain-specific classification of Judgments.

2 Related Work

In [1], they have considered the Clustering of Indian legal judgments by exploiting the features such as citation and paragraph link. In future work, they have mentioned using Acts as a feature for clustering. This paper is considered the baseline for our study.

Whereas the authors in [3] used LDA-based Topic modelling for clustering Indian legal data. Then a similarity measure is applied.

In [2], they have used data from the Special Civil Court located at the Federal University of Santa Catarina. They applied different clustering algorithms and found that Hierarchical clustering performed better. The Judgments were from the domain of Consumer Law, and they wanted cases in which consumers claimed moral and material compensation from airlines for service failures.

The authors in [4] used embedding-based representation for representing the documents and paragraphs. Then Nearest Neighbor Search is used to retrieve most similar paragraphs. Their approach was able to identify documents of similar topics even with noisy data.

The paper [5] used hierarchical LDA to find topics, and similarity measures are applied to legal judgments for creating the summary.

In paper [6], they have used an iterative Latent Semantic Analysis (LSA)-based concept for clustering.

In paper [7], they used closed lawsuits in Chinese legal documents as training data. They have used keyword-based and case-based data. They have created those data with NLP technologies.

The authors in paper [8] used a classification-based recursive soft clustering algorithm with built-in topic segmentation. They used topical classifications, document citations, and click stream data as features to create the cluster model.

3 Proposed System

The existing approaches for similar judgment analysis and clustering used different features. The recent work by [9] used embedding mechanisms for finding

Document similarity, which gave good results for the whole document than citations and paragraph links. Using the whole document for clustering is a good approach if no other relevant features are available. For searching and retrieval, avoiding lengthy information as features makes the search process cost-efficient and time-efficient. So the study was to find the relevant feature for efficient domain-specific classification of Indian legal judgments. Here the information such as Acts mentioned in the Judgment, Acts with section, Citations present in the judgment, and the whole judgment is considered for the experiment.

In [1], they initially considered citations only for cluster analysis. However, they found considerable judgment needs more citations when analyzing real-world data. They used paragraph links with the judgment to perform clustering to increase the number of citations.

None of the previous works tried the feature Act for Clustering as no dataset is available for Judgments with their associated Acts for Indian legal data. Here the information such as Acts, Acts with Sections, and Citation information is extracted using a regular expression-based compiler for each judgment.

3.1 Feature selection or extraction

Finding the best feature to form a cluster.

Acts- An Act is a written rule by the legislative authority(Wikipedia). The lawyers justify their opinion by pointing out the existing rules or acts. There will be some legal Judgments that may only contain precedents and not statutes. Most of the Acts in India use keywords based on the domain. The Act names for Land-related Judgments include Land Acquisition Act, 1894, Right to Fair Compensation and Transparency in Land Acquisition, Rehabilitation and Resettlement Act, 2013, similarly for Tax related Income Tax Acts, etc. The name of the Acts in the judgment gives an idea about the Case Category. Also, if some cases contain both Land Acquisition and Income Tax Act, then lawyers can easily conclude that the case must be related to the compensation amount after land acquisition. If more than one case contains these two acts, then there is a high probability of having similar facts for these Judgments. A judgment may refer to many acts based on a legal dispute.

Acts with sections: Acts with Sections (Section 2 of the Land Acquisition Act, 2013) can also be used as a feature for clustering. However, it is more appropriate for making sub-clusters. A judgment dataset of a year with 500 Judgments will rarely contain similar sections. Also, different land-based cases use different sections. So there may not be a chance of forming a cluster using that information. However, it can be a good feature in forming a sub-cluster or if dataset is very huge. To analyze the validity of the assumption that feature is also considered for the study.

Citation Information - Legal citations, in general, are used to identify the source of information supporting a particular point in a legal document [18]. The citation information will be very useful for legal disputes based on common law. However, clustering using legal citation information contains many difficulties. Different judges use different citations(A judgment may be present in more than

one journal; Different journals will have different citation numbers and styles). The Supreme Court of India has taken the initiative to standardize the citation. However, the old judgments, the primary data source for analysis, follows that pattern.

Also, as discussed in [1], fewer judgments have similar citations. So, the citation information cannot lead to a better cluster, especially when the dataset size is very small. It can give better results with huge datasets where all judgments are available.

Whole Document - Considering the whole document is a better approach for finding [9] similarities between some legal documents but not for clustering huge data. Also, a single judgment may discuss multiple issues[9]. The advanced transformer architectures can give better document embedding vectors. However, generating embeddings for lengthy legal documents, especially for huge datasets, might be time-consuming. Also, some of the characteristics of the Indian judgment, such as length, Unstructured pattern, the presence of headers and footers, etc., lead to confusion even in the advanced embedding approaches.

3.2 Clustering algorithm selection

Selecting the best algorithm suitable for the data in hand.

For clustering, popular clustering algorithms were chosen. Choosing the best clustering algorithm depends on many factors, such as the nature of the data, expected output, and dataset size. The algorithm that best suits our problem must be found from the existing clustering algorithms.

Some of the popular clustering algorithms selected for the study include the algorithms that do not want to specify the number of clusters. The short-listed algorithms include Affinity Propagation[13], DBSCAN [12], OPTICS[14], Mean Shift[15], and HDBSCAN[16]. The algorithms have their own advantages and disadvantages. For some algorithms, the number of clusters needs to be specified; for some algorithms, the damping coefficient must be specified, etc.

4 Experiments and Evaluation

Due to the unavailability of the gold standard dataset, evaluation is done on a single cluster (Income Tax Cluster), and the purity of the single cluster is evaluated.

Preprocessing Before applying the clustering technique, certain preprocessing is done on the dataset, as the dataset chosen does not contain any labelled data. The 2020LegalJudgments are collected through web scraping from the LIIofIndia website. The LIIofIndia website contains raw judgments directly from the court with different headers and footers. So the headers and footers in the PDF must be cleaned to make the dataset. Finding a generalized approach to clean every judgment headers and footers was a challenging task.

Information extraction As the study is on finding the best feature for legal judgment clustering, some features must be extracted from the judgment. The features such as Acts, Acts with sections, and Citations are extracted from each judgment. These features are extracted using a regular expression-based compiler. Extracting information was also challenging, as the Indian legal system does not follow a structured writing pattern. Acts extraction was very challenging as it was not easy to find the boundaries of the Act. The Act's length and nature seemed like a sentence, and it was confusing. Making a generalized pattern for Act extraction was difficult.

Vectorization The information is in the form of natural language. The Judgment, Acts, Acts with section information, Citations, everything is in natural language. This needs to be converted into vectors for applying a cluster model. For vectorization, the pre-trained Sentence Transformer BERT[17] seems more effective than the traditional embedding techniques as it can derive meaningful embeddings incorporating the semantic features of the text. Four models of Sentence Transformer BERT are applied to convert every feature into corresponding vectors. The legal sentences are lengthy and confusing. Not just legal sentences, the other features, such as Acts and Acts with sections, are also lengthy. So Sentence Transformer can work well with every feature chosen here for the study.

4.1 Experiments

Choosing a suitable Sentence Transformer Embedding Embeddings are created for all four features using four different embedding models. Four sentence transformer embeddings are chosen bert-base-nli-mean-tokens, all-mpnet-base-v2, all-MiniLM-L6-v2 and paraphrase-mpnet-base-v2 [19]. The bert-base-nli-mean-tokens is the basic model, whereas all-mpnet-base-v2 is the model that gives the best quality results. The all-MiniLM-L6-v2 is a very fast model that gives the best results, and paraphrase-mpnet-base-v2 trained on the multilingual dataset is used to treat legal language as a new language as the characteristics of legal language are different. This was the multilingual model that gave good results in less time. The time taken for embedding different features of the 2020LegalJudgments with 550 data points is recorded. The time taken by different pre-trained embeddings to create the embedding of different features chosen for study is recorded in Table 1.

Table 1. Time taken by different embeddings on different features (in mins)

Embedding Chosen	Acts	Judgments	Act with section	citations
bert-base-nli-mean-tokens	0.472	1.915	0.929	0.720
all-mpnet-base-v2	0.771	7.618	1.585	1.262
all-MiniLM-L6-v2	0.096	0.950	0.234	0.171
paraphrase-mpnet-base-v2	0.915	10.342	10791	1.517

The feature 'Judgment' is lengthy, so the time taken to create an embedding for the judgment was very high. The feature 'Acts' took very less time among the four features. As different features are of different lengths, an additional analysis of the performance of the four embeddings on the clustering result should also be considered. all-MiniLM-L6-v2 embedding took less time for all features and bert-base-nli-mean-tokens is on second place .

Choosing a suitable algorithm Based on the nature of the problem, where there is no information about the number of clusters needed, the algorithms that do not require the number of clusters are chosen for study. The Affinity Propagation, DBSCAN, OPTICS, HDBSCAN, and Mean Shift are the popular algorithms that do not require the number of clusters to be specified. The results of the four algorithms, except Meanshift, are recorded. The Mean Shift algorithm did not converge at all. For analysis, a cluster is chosen from the 2020Judgment-Dataset. The data points that contain the *Prevention of Corruption Act* are filtered [6, 58, 107, 166, 290, 302, 324, 332, 350, 413, 485, 502, 508]. There were only two data points in the pure cluster([502,58] - Prevention of Corruption Act) containing only the Prevention of Corruption Act. Performance of different algorithms with different embeddings chosen are recorded in Table 2, 3, 4, 5 below.

Table 2. Affinity Propagation: Comparing the results of algorithms chosen

Embedding	Converges at	Cluster size	Cluster points	Cluster number
bert-base-nli-mean-tokens	13	41	(58,502)	34
all-mpnet-base-v2	27	39	(58,166,417,502)	3
all-MiniLM-L6-v2	14	42	(58,166,502)	1
paraphrase-mpnet-base-v2	13	41	(58,166,417,502)	1

Table 3. DBSCAN: Comparing the results of algorithms chosen

Embedding	Epsilon and minsamples	Cluster size	Cluster points	Cluster number
bert-base-nli-mean-tokens	(0.5,2)	22	(58,502)	6
all-mpnet-base-v2	(0.5,2)	36	(58,502)	6
all-MiniLM-L6-v2	(0.5,2)	48	(58,502)	10
paraphrase-mpnet-base-v2	(0.5,2)	41	(58,502))	6

The algorithms that gave better results with minimum time are chosen for the next step. The Affinity Propagation algorithm with bert-base-nli-mean-tokens gave the pure cluster. With all-MiniLM-L6-v2 embedding also, the Affinity Propagation algorithm gave better results. For the other two embeddings, the noise

Table 4. OPTICS: Comparing the results of algorithms chosen

Embedding	Epsilon and minsamples	Cluster size	Cluster points	Cluster number
bert-base-nli-mean-tokens	(0.5,2)	67	(58,502)	62
all-mpnet-base-v2	(0.5,2)	74	(58,502)	31
all-MiniLM-L6-v2	(0.5,2)	69	(58,502)	57
paraphrase-mpnet-base-v2	(0.5,2)	66	(58,502))	42

Table 5. HDBSCAN: Comparing the results of algorithms chosen

Embedding	minclustersize	Cluster size	Cluster points	Cluster number
bert-base-nli-mean-tokens	(2)	83	(58,502)	3
all-mpnet-base-v2	(2)	87	(58,502)	49
all-MiniLM-L6-v2	(2)	81	(58,502)	15
paraphrase-mpnet-base-v2	(2)	85	(58,502))	27

was there in the result. Also, for different embeddings, the algorithm converged at different times, and finding the converging point of the algorithm is time-consuming.

Moving on to the DBSCAN algorithm, it gave good results in all four embeddings keeping the epsilon = 0.5 and min samples = 2, where epsilon is the minimum distance between two samples and min samples is the number of samples in a neighbourhood for a point to be considered as a core point. The number of clusters was different for the four embeddings.

The algorithm OPTICS is an improvised version of the DBSCAN algorithm without being sensitive to the radius of the neighbourhood and the number of points in the neighbourhood[10]. Keeping the epsilon and min samples parameters the same in DBSCAN, OPTICS gave more clusters than DBSCAN. On analyzing the results manually, the increased number of clusters in OPTICS reduced the unclassified points, and the purity of the cluster was also good.

The last algorithm chosen for the study is HDBSCAN which requires only one compulsory parameter. i.e., minclustersize, the minimum number of elements required to form a cluster. Setting the minclustersize = 2, the algorithm is run with different embeddings. The results were good, and the number of clusters was higher than all four algorithms which gave more pure clusters.

OPTICS and HDBSCAN gave good results with minimum time among the four algorithms. From the literature, both algorithms' time complexity is considered O(nlogn). However, the HDBSCAN ran faster than OPTICS. For clustering with Acts using paraphrase-mpnet-base-v2, the HDBSCAN took only 0.009 minutes, whereas OPTICS took 0.0793 minutes. HDBSCAN also worked better in terms of the number of unclassified data points and the number of clusters.

Density-based algorithms are said to be very useful for datasets of arbitrary shape[10]. The vector embeddings projected onto vector space were also accurate for creating a good density-based partition. Density-based algorithms gave a better result for arbitrary shape data with the help of high-quality embeddings.

HDBSCAN, a combination of Hierarchical and Density-based, gave better results for the same reason with minimum time. The DBSCAN looks for uniform-density clusters, whereas the HDBSCAN looks for varying-density clusters [20], resulting in more clusters with high purity.

Choosing the best feature

The table 6 below shows the performance of different features with HDBSCAN on the dataset. The fourth column shows the data points and the cluster number. Around 24 [**6, 191, 243, 245, 253, 255, 266, 296, 304, 329, 340, 341, 342, 355, 357, 389,419, 424, 428, 432, 440, 461, 503, 510**] judgment entries are there with Income Tax Act. But in some cases, the Income tax will have less priority as some other prominent acts must indicate the case category. Here, the income tax act with the Companies Act or Finance Act is chosen for the study. But it will be in different clusters for different features and different embeddings. So the most prominent Income Tax clusters formed are recorded along with cluster numbers. A highly pure cluster(the common Acts will be the same for all cluster members) is chosen for feature selection as it can properly define the unclassified data points, which is an important feature for cluster evaluation. For real-life systems, choosing the high value as minclustersize will be good.

Moving on to finding the best feature for Clustering, Acts, and Acts with section seems to have given good results with minimum unclassified points with high purity cluster. The judgments also performed well but not as good as the Acts. For the feature citations' unclassified points are much less, but cluster purity was very low. The citations did not give better results; it may be due to the extraction quality of the citation. For citations, just the name of the citation ((1996) SCC 12) is extracted and not the party names. Some judgments also give the party names and citations, while others do not. As it can lead to more confusion in assigning the party names and the associated citations using automatic information extraction, just the citations are extracted.

On evaluating the quality of clusters manually, the *citation feature* created a lot of clusters but not of good quality. As the citation information is too short, the citation embeddings with a similar year or number are projected into a similar vector space, and hence clusters are formed. Data points 253 and 296 are the common data points present in clustering using the Judgments, Acts and Acts with sections. But those points are not there in the citation-based clustering result.

The *Judgments feature* also performed well in forming clusters. But the cluster quality is not as good as those formed with Acts. Acts with sections gave good results in grouping clusters. But as discussed earlier, the sections will be unique, and there were not enough similar sections to form a group as a cluster as the dataset size is only 550. Here, a dataset for the year 2020 is considered, as most of the clustering applications will be for clustering the judgments every year. Even with low data points, the clustering with the feature Act gave good results.

For a classification kind of result, the *Acts feature* is the best. The main advantage of using Act as feature is that searching using the keyword or Act name reduces the search space in the practical scenario than comparing whole judgments. Using the Act feature results in good results with minimum time. The number of unclassified data points is low in the case of the Acts feature with higher cluster quality. The disadvantage of using this feature is that additional time is needed to extract the Act information. Also, there may be some judgments without any Acts. Those Judgments without Act information remain unclassified, or those which do not have acts will form another cluster.

The *Acts with section* will group the Judgments only if the same section is in the other judgment. So, the number of unclassified data points is higher in this case. As our assumption, the Acts with section information can be used to create subclusters or will be suitable for large datasets.

	acts_wof	Journal
306	'Disaster Management Act , 2005', 'Disaster Management (National Disaster Response Fo...	INSC 316
397	'Disaster Management Act , 2005', 'Industrial Disputes Act , 1947', 'Section of Disast...	INSC 408
412	'Disaster Management Act , 2005'	INSC 423
461	'Disaster Management Act , 2005', 'Act , 2005', 'Companies Act , 2013', 'Income Tax Ac...	INSC 473

Fig. 1. Results showing the embedding is not clustering on the basis of year

The advantage of using whole Judgments is that they can be used without extracting any information. Searching or comparing two Judgments in a cluster can be hectic. Also, the time taken for embedding is high, and cluster quality is low compared to the Acts feature.

Another important insight from the study is about embedding. The embedding bert-base-nli-mean-tokens, all-MiniLM-L6-v2 took less time for creating an embedding all the features and performed well. For Whole Judgments, paraphrase-mpnet-base-v2 gave a good quality cluster with more data points. The reason for the good results of paraphrase-mpnet-base-v2 on Judgments than the other three is because of the characteristics of legal sentences and as it is trained on a multilingual dataset.

4.2 Evaluation

From the above experiments, the best feature, best sentence transformer embedding, and best clustering method are found considering the practical applicability. The HDBSCAN algorithm with minimum cluster size is chosen to find the best feature for clustering the legal judgment. The judgments containing the Income Tax Act are chosen to evaluate the quality of the cluster. The quality of each feature is evaluated by Act information of the judgment. For evaluating the clusters based on each feature, the number of judgments with common acts present in the cluster and the total number of judgments present in the cluster are used. The number of judgments with common acts divided by the total number of judgments in the cluster can evaluate the purity of the individual cluster

Table 6. Comparison of performance of different features with HDBSCAN

Feature	Embedding	Cluster size	Unclassified	Cluster points& cluster
Judgment	bert-base-nli-mean-tokens	92	307	(253,296,472-**39**) (329,440-**40**)
	all-mpnet-base-v2	80	293	(253,296,329,440-**7**)
	all-MiniLM-L6-v2	72	338	(130,253,296,329,440,472-**27**)
	paraphrase-mpnet-base-v2	79	335	(130,253,296,329,389,419,440-**23**)
Acts	bert-base-nli-mean-tokens	83	143	(191,255,266,342-**49**) (56,345,428,503-**50**) (243,253,296,304,329,340,341,389-**51**) (432,510-**52**)
	all-mpnet-base-v2	87	155	173,296,340,341,357,389,399,432, 445,472,503,510,527-**79**) (191,255,266,342-**23**) (243,253,304-**74**) (329,355-**69**) (56,345,419-**64**)
	all-MiniLM-L6-v2	81	151	(56,345,419,428-**53**) (243,253,296,304,340,389,510-**60**) (191,255,266,342,355-**61**)
	paraphrase-mpnet-base-v2	85	157	(243,253,304-**73**) (296,340,341,389,510-**72**) (191,255,266,342,355-**20**)
Acts with section	bert-base-nli-mean-tokens	77	248	(243,255,296,329,341,342, 355,389,432,510-**49**)
	all-mpnet-base-v2	85	203	(243,253,255,296,329,357,432-**48**)
	all-MiniLM-L6-v2	91	202	(56,130,167,191,243,253,255,262, 296,329,345,355,432,467,490,510-**56**)
	paraphrase-mpnet-base-v2	87	249	(173,253,304,340,421,522-**57**)
Citations	bert-base-nli-mean-tokens	21	110	(440,510-**10**)
	all-mpnet-base-v2	61	272	(341,355,357,432-**30)**
	all-MiniLM-L6-v2	56	292	(432,472-**34**)
	paraphrase-mpnet-base-v2	53	300	(341,355,357,389,428,432,472,503-**18**)

formed. Purity value 1(100 %) indicates a highly pure cluster, and purity value 0 indicates a highly impure cluster.

$$Purity\ of\ the\ cluster\ i\ = A/B \tag{1}$$

A = Number of data points with prominent Act present in the cluster i
B = Total number of data points in the cluster i
E.g., purity of the cluster i = number of data points with Income Tax Acts present in the cluster i/Total number of data points in the cluster i

As most of the practical applications for legal judgment clustering come with unknown cluster numbers, it is found that HDBSCAN will perform better not just in terms of quality but in case of time efficiency also. When choosing models for legal judgments, time efficiency is very important. As the Judgments are very large, consisting of 25 pages on average, it will take a lot of time to load a big dataset. Usually, clustering applications may contain a lot of data. So the clustering methods should be time efficient to perform better on a huge dataset. Also, Acts is considered the best feature, with minimum unclassified points and a high-quality cluster. all-MiniLM-L6-v2 is considered the best embedding, which takes very less time for encoding compared to other encoders while maintaining the quality.

The clustering quality of the HDBSCAN was very pure, which lead us to think whether clustering is done based on the year information present at the end of the Act. If that is true, then [Right to Information Act,2005] and [Disaster Management Act, 2005] should fall under the same cluster. But both were in different clusters as in Figure 1. Also, some noises are added to the year information present in the list of unique Acts. The results show that the clustering is not done based on the year information present at the end of the Act.

The HDBSCAN algorithm is run with all-MiniLM-L6-v2 embedding and Acts as a feature by varying the minimum cluster size(minimum number of data points in a cluster)and keeping minimum number of samples (minimum number of data points that should be present in the nearby region to form a cluster) as two. The purity and unclassified data points are recorded in Table7 and depicted in Figure 2. The results shows that the HDBSCAN algorithm got converged at minclustersize= 6 with 19 clusters while maintaining (19/21)90% purity. Among 24 Income Tax points, 19 of them came under a single cluster, and 5 of them remained as unclassified.

	acts_wof	Journal
56	'Central Sales Tax Act , 1956', 'Customs Act , 1962', 'General Sales Tax Act , 1959', ...	INSC 60
191	'Income Tax Act , 1961'	INSC 198
243	'Companies Act , 1956', 'Income Tax Act , 1961', 'Income Tax Rules , 1962'	INSC 250
253	'Income Tax Act , 1961', 'Companies Act , 1956'	INSC 261
255	'Income Tax Act , 1961'	INSC 263
266	'Income Tax Act , 1961'	INSC 275
296	'Income Tax Act , 1961', 'Finance Act , 1993', 'Taxation Laws (Amendment) Act , 1991...	INSC 305
304	'Income Tax Act , 1961', 'Companies Act , 1956', 'Indian Companies Act , 1956'	INSC 314
329	'Direct Tax Laws (Amendment) Act , 1987', 'Income Tax Act , 1961'	INSC 339
340	'Finance Act , 2001', 'Companies Act , 1956', 'Finance Bill , 1983', 'Income Tax Act ,...	INSC 350
341	'Foreign Exchange Regulation Act , 1973', 'Income-tax Act , 1961', 'Income Tax Act , 1...	INSC 351
342	'Income Tax Act , 1961'	INSC 352
345	'Act , 1956', 'Rajasthan Sales Tax Act , 1954', 'Central Sales Tax Act , 1956', 'Finan...	INSC 355
355	'Income Tax Act , 1961', 'By Finance Act , 2012'	INSC 365
357	'Income Tax Act , 1961', 'Finance Act , 2016', 'Finance Act , 2012', 'Finance Act , 20...	INSC 367
389	'Income-tax Act , 1961', 'Finance (No.2) Act , 1967', 'Central Excise Tariff Act , 1...	INSC 400
419	'Central Sales Tax Act , 1956', 'Tamil Nadu Indian Made Foreign Spirits (Manufacture ...	INSC 430
428	'Central Sales Tax Act , 1956', 'Rajasthan Value Added Tax Act , 2003', 'Rajasthan Sal...	INSC 439
432	'Income Tax Act , 1961', 'Finance Act , 2002', 'Code of Civil Procedure , 1908'	INSC 443
503	'Indian Income Tax Act , 1922', 'Finance Act , 1981', 'Societies Registration Act , 18...	INSC 516
510	'Income Tax Act , 1961', 'Finance Act , 2003', 'Finance Act , 2001', 'Finance Act , 20...	INSC 524

Fig. 2. Result of converged cluster(Income Tax Act) at minclustersize=6

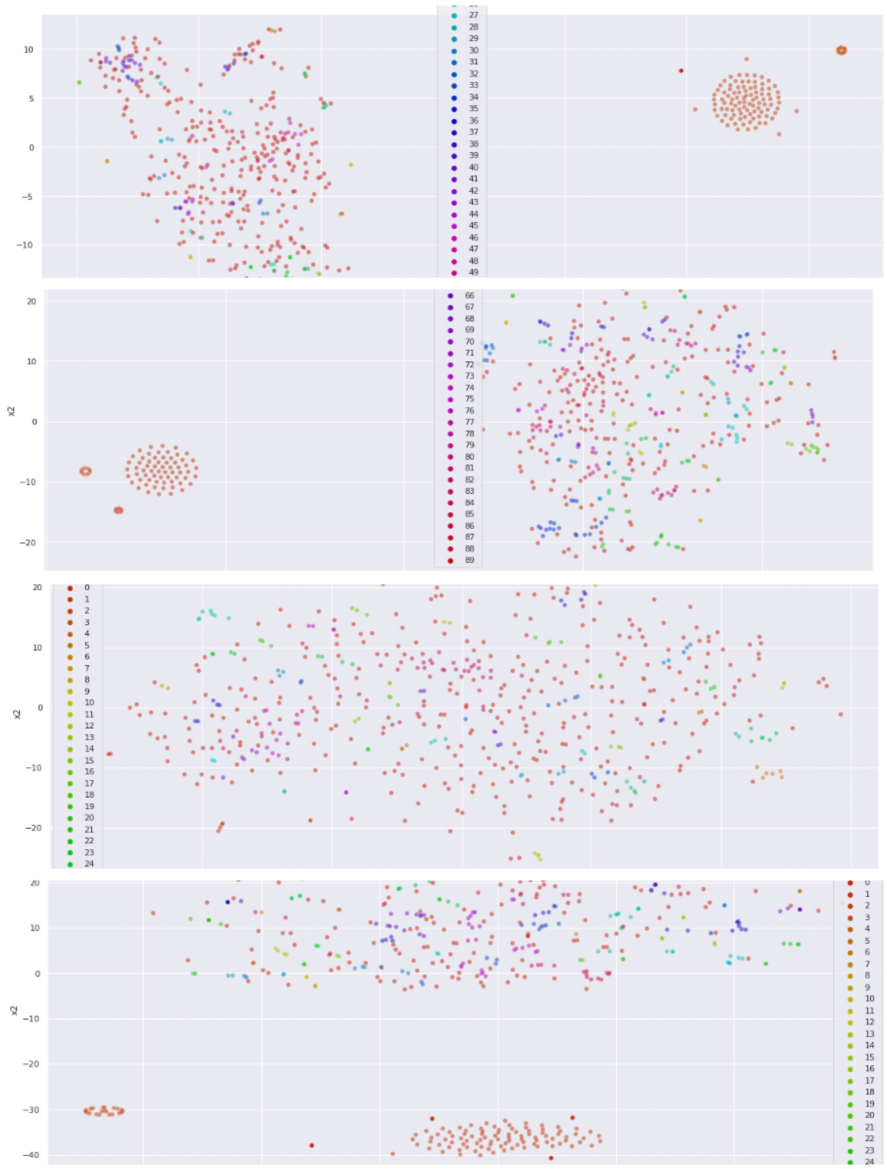


Fig. 3. Clustering plots of four different features(Citations , Act with sections , Judgment, Acts)

Table 7. HDBSCAN: Finding the convergence

min cluster size	No: of clusters formed	No: of Unclassified points	Clusters with Income Tax points Cluster number : Income Tax data points in the cluster	Total no: of income tax points	Purity Incometax points /total points
2	81	151	38 - [6]	1	1/3
			61 - [191,255,266,342,355]	5	5/5 =100%
			60 - [243,253,296,304,340,389,510]	7	7/7 =100%
			53 - [419,428]	2	2/4
			26 - [461]	1	1/4
			-1 - [245,329,341,357,424,432,440,503]	8	8/151
3	42	214	15 - [6]	1	1/3
			30 - [191,255,266,342,355]	5	5/5 =100%
			29 - [243,253,296,304,340,389,510]	7	7/7 =100%
			22 - [419,428]	2	2/4
			8 - [461]	1	1/4
			-1 - [245,329,341,357,424,432,440,503]	8	8/214
4	29	211	23 - [191,255,266,342,355]	5	5/5 =100%
			22 - [243,253,296,304,340,389,510]	7	7/7 =100%
			17 - [419,428]	2	2/4
			7 - [461]	1	1/4
			-1 - [6,245,329,341,357,424,432,440,503]	9	9/211
5	20	219	12 - [191,243,253,255,266,296,304,329,340, 341,342,355,357,389,419,428,432,503,510]	19	19/21 =90%
			-1 - [6,245,424,440,461]	5	5/219
6	**19**	**224**	**11 - [191,243,253,255,266,296,304,329,340, 341,342,355,357,389,419,428,432,503,510]**	**19**	**19/21 =90%**
			-1 - [6,245,424,440,461]	5	5/224
7	4	6	0 - [6 ,191, 243,245,253,255, 266, 296, 304, 329,340, 341, 342, 355, 357, 389, 419, 424, 428, 432, 440, 461, 503, 510]	24	24/395 = 6.07%

5 Conclusion

In this paper, few experiments were conducted to find the best feature suitable for clustering legal data. There were prior works exploiting Judgment, Citations, and Citations with paragraph links, Topic modeling to cluster the legal information. Here, the Acts, Acts with Sec, Judgment, and Citation information is considered. The evaluation shows that the Acts information gives a best result than the other three. For all-MiniLM-L6-v2 embedding with HDBSCAN algorithm the cluster purity was around 90%. In future work, we would like to exploit the important keywords or Topics along with the Act information.

References

1. Raghav, K., et al. "Text and citations based cluster analysis of legal judgments." Mining Intelligence and Knowledge Exploration: Third International Conference,

MIKE 2015, Hyderabad, India, December 9-11, 2015, Proceedings 3. Springer International Publishing, 2015.
2. Sabo, Isabela Cristina, et al. "Clustering of Brazilian legal judgments about failures in air transport service: an evaluation of different approaches." Artificial Intelligence and Law 30.1 (2022): 21-57.
3. Raghuveer, K. "Legal documents clustering using latent dirichlet allocation." International Journal of Applied Information Systems 2.1 (2012): 34-37.
4. De Martino, Graziella, Gianvito Pio, and Michelangelo Ceci. "PRILJ: an efficient two-step method based on embedding and clustering for the identification of regularities in legal case judgments." Artificial Intelligence and Law 30.3 (2022): 359-390.
5. Venkatesh, Ravi Kumar. "Legal documents clustering and summarization using hierarchical latent Dirichlet allocation." IAES International Journal of Artificial Intelligence 2.1 (2013).
6. Mathai, Sumi, Deepa Gupta, and G. Radhakrishnan. "Iterative concept-based clustering of Indian Court Judgments." Proceedings of the Second International Conference on Computational Intelligence and Informatics: ICCII 2017. Springer Singapore, 2018.
7. Liu, Chao-Lin, Cheng-Tsung Chang, and Jim-How Ho. "Classification and clustering for case-based criminal summary judgments." Proceedings of the 9th international conference on Artificial intelligence and law. 2003.
8. Lu, Qiang, et al. "Legal document clustering with built-in topic segmentation." Proceedings of the 20th ACM international conference on Information and knowledge management. 2011.
9. Mandal, Arpan, et al. "Measuring similarity among legal court case documents." Proceedings of the 10th annual ACM India compute conference. 2017.
10. Xu, Rui, and Donald Wunsch. "Survey of clustering algorithms." IEEE Transactions on neural networks 16.3 (2005): 645-678.
11. Van Opijnen, Marc, and Cristiana Santos. "On the concept of relevance in legal information retrieval." Artificial Intelligence and Law 25 (2017): 65-87.
12. M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining (KDD'96), 1996, pp. 226–231.
13. Frey, Brendan J., and Delbert Dueck. "Clustering by passing messages between data points." science 315.5814 (2007): 972-976.
14. Ankerst, Mihael, et al. "Ordering points to identify the clustering structure." Proc. ACM SIGMOD. Vol. 99. 2008
15. Cheng, Yizong. "Mean shift, mode seeking, and clustering." IEEE transactions on pattern analysis and machine intelligence 17.8 (1995): 790-799.
16. McInnes, Leland, John Healy, and Steve Astels. "hdbscan: Hierarchical density based clustering." J. Open Source Softw. 2.11 (2017): 205.
17. Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." arXiv preprint arXiv:1908.10084 (2019).
18. https://guides.loc.gov/case-law/citations. Last accessed 17 April 2023
19. https://www.sbert.net/docs/pretrained_models.html. Last accessed 17 April 2023
20. https://towardsdatascience.com/density-based-clustering-dbscan-vs-hdbscan-39e02af990c7. Last accessed 17 April 2023
21. Kumar, Sushanta, et al. "Finding similar legal judgements under common law system." Databases in Networked Information Systems: 8th International Workshop, DNIS 2013, Aizu-Wakamatsu, Japan, March 25-27, 2013. Proceedings 8. Springer Berlin Heidelberg, 2013.

Citation Recommendation on Scholarly Legal Articles

Doğukan Arslan[1][0000−0002−8114−2953], Saadet Sena Erdoğan[2][0000−0001−5304−3483], and Gülşen Eryiğit[1][0000−0003−4607−7305]

[1] Department of Artificial Intelligence and Data Engineering
Istanbul Technical University, İstanbul, Türkiye
{arslan.dogukan,gulsenc}@itu.edu.tr
[2] Department of Computer Engineering
Istanbul Technical University, İstanbul, Türkiye
erdogansa20@itu.edu.tr

Abstract. Citation recommendation is the task of finding appropriate citations based on a given piece of text. The proposed datasets for this task consist mainly of several scientific fields, lacking some core ones, such as law. Furthermore, citation recommendation is used within the legal domain to identify supporting arguments, utilizing non-scholarly legal articles. In order to alleviate the limitations of existing studies, we gather the first scholarly legal dataset for the task of citation recommendation. Also, we conduct experiments with state-of-the-art models and compare their performance on this dataset. The study suggests that, while BM25 is a strong benchmark for the legal citation recommendation task, the most effective method involves implementing a two-step process that entails pre-fetching with BM25, followed by re-ranking with SciNCL or SGPT, which enhances the performance of the baseline from 0.26 to 0.29 MAP@10. Moreover, fine-tuning leads to considerable performance increases in pre-trained models, which shows the importance of including legal articles in the training data of these models.

Keywords: Citation Recommendation · Legal Natural Language Processing · Information Retrieval

1 Introduction

The task of citation recommendation involves identifying potential citations among some candidates for a particular text, specifically for justifying arguments or making concepts clear. Predominantly, studies in this task neglect the lack of diversity and the imbalance of datasets regarding article fields, which might affect the performance of models. The challenge of this issue has been recently addressed in [39], proposing a new benchmark dataset that spans diverse scientific fields along with the field-level evaluation of various models. Nevertheless, certain core fields, such as law, remain excluded.

In the context of legal natural language processing, citation recommendation is primarily used to discover justifying arguments from non-scholarly law

related articles, mostly judicial opinions [24,17,58], which leaves scholarly legal articles unexplored. Since it is possible to automatically generate labeled data for the task of citation recommendation, models trained on scholarly legal articles could offer considerable value for various legal natural language processing tasks, including legal case retrieval [31], legal document similarity [8], and legal judgement prediction [16].

In order to address this, we introduce the very first scholarly legal citation recommendation dataset in the literature: We gather 719 scholarly legal articles with 10K incoming citation links from 9K articles. Additionally, this paper provides baseline scores for the citation recommendation task on scholarly legal articles using state-of-the-art models such as BM25 [53], Law2Vec [14], SciBERT [6], SPECTER [15], LegalBERT [13], SciNCL [46], and SGPT [42] in four different setups: (i) as a baseline, a BM25 model is trained with the gathered dataset. The performance of this model in finding relevant articles to cite is evaluated along with the aforementioned pre-trained models. (ii) pre-trained models are fine-tuned using the gathered dataset. Next, the task is divided into two: pre-fetching and re-ranking. Then, (iii) pre-trained and (iv) fine-tuned models are utilized for re-ranking articles retrieved by BM25. We demonstrate that BM25 is a suitable baseline approach with competitive results for citation recommendation on scholarly legal articles. Overall, a two-step approach consisting of pre-fetching with BM25 and subsequent re-ranking with SciNCL or SGPT performs most effectively on our dataset. Besides, the performance of the fine-tuned models in comparison to the pre-trained ones, with a small-sized dataset, indicates that models trained with scholarly texts should include articles from more varied scientific fields, including the legal domain. The dataset and fully reproducible code is publicly available on GitHub[3].

The remainder of this paper is structured as follows. Section 2 gives a general introduction to citation recommendation methods and how they are applied to the legal domain. Section 3 includes considered approaches. The dataset and experimental setup, along with the evaluation metrics and obtained results, are presented and discussed in Section 4, and Section 5 contains final thoughts and suggestions for future studies.

2 Related Work

Parallel to the rapid growth in scientific publication activity [10], recent papers have more outgoing citations than before [59]. This issue raises concerns about the quality of citations. Hence, the task of citation recommendation became popular. Consequently, the citation recommendation task has been studied with documents from different domains eg., patents [35], Wikipedia articles [22], news [47] and legal cases [24]. In this section, we examine citation recommendation studies from a broader perspective and discuss their implications to the law field. One may refer to [20,32,38] for detailed surveys regarding citation recommendation studies.

[3] https://github.com/dgknrsln/LegalCitationRecommendation

2.1 Citation Recommendation

Practiced methods in citation recommendation (CR) can be listed as collaborative filtering (CF), graph-based filtering (GB), and content-based filtering (CB). CF algorithms aims to match different users' preferences to recommend an article [48,37,29]. This kind of approaches can be brittle, particularly when data is sparse or when a newly included item has few or no evaluations from users (early-rater problem), or when a new user that the system has no knowledge of joins the system (cold-start problem) [2]. In GB algorithms, a recommender system is modeled using a graph and relations between authors, papers, venues etc. [26,4,28]. They may encounter computational complexity issues [2] and the problem of bias against old nodes in the network [1]. CB algorithms take advantage of papers' descriptive features (content such as title, abstract, sentences, or key-phrases) for CR [7,27,3]. Attaining those features is easy, and most recommendation systems rely on a CB method [5], as in this study.

CR studies can roughly be categorized as local and global, concerning the extent of the recommendation. In *local citation recommendation* (LCR), also called *context-aware citation recommendation*, the primary focus is a specific part of the input document, such as a sentence or a slightly larger window, whereas, in *global citation recommendation* (GCR), the entire document [43] or its abstract [33,41,62] is considered. Different types of background knowledge may also be utilized in GCR studies including, but not limited to the title [7,23,25], author information [23,34,25], venue [52,62,63], and key-phrases [33,34,41].

Various scholarly datasets have been collected regarding training and evaluation of the CR methods. While most of the datasets consist of papers from computer science and related fields with citation links and metadata information, such as DBLP [57], ACL-AAN [49], ACL-ARC [9], arXiv CS [21], Scholarly Dataset [56], and unarXiv [55], some includes papers from medicine like PubMed[4] and RELISH [12] or from a wide variety of fields as CORE[5], S2ORC [30], CiteSeer[6], and MDCR [39]. However, none of them include articles from the field of law.

2.2 Legal Citation Recommendation

CR methods are applied to the legal domain, focusing mainly on finding appropriate non-scholarly legal documents to cite, such as court decisions, statutes, and law articles. One of the early works in legal CR makes use of a CF approach to build a legal recommender system [60]. [58] tries to solve the legal document recommendation problem, studying it in a CR context and utilizing citation network analysis and clickstream analysis. [45] studies various citation-based graph methods for the legal document recommendation task, representing documents as nodes and citation links as edges in the graph. Varying CB and CF methods are applied in [24] to find proper legal documents to cite. Evaluations on several

[4] https://www.nlm.nih.gov/databases/download/pubmed_medline.html
[5] https://core.ac.uk/documentation/dataset
[6] https://csxstatic.ist.psu.edu/downloads/data

metrics demonstrate the strength of neural models over traditional ones in the legal CR task. [19] works with legal knowledge graphs to predict citation links and find similar cases.

3 Methods

The following seven approaches are utilized for the legal CR task in four different setups, which are applying pre-trained language models directly, fine-tuning pre-trained models before using, and first pre-fetching articles with BM25, then utilizing pre-trained and fine-tuned models to re-rank the retrieved results. This multi-stage approach is often employed in information retrieval systems [44], where faster but less accurate models like BM25 are used in the initial stage to retrieve a subset of related documents. In the subsequent stage, slower but more precise models refine the ranking of the top candidate list to improve the retrieval system's effectiveness. This separation of retrieval into stages enables the retrieval system to achieve a good trade-off between efficiency and effectiveness.

BM25 Given a query, Best Matching-25 scores the relevance of documents. It is a strong baseline for legal case retrieval task [54], which is quite similar with citation recomendation. Okapi BM25 implementation of a Python package called rank-bm25 [11] is trained with the abstracts of LawArXiv articles. Then, the 10 most relevant articles are retrieved for each query article (∼9K). This model is also used in the pre-fetching step as in traditional information retrieval studies.

Law2Vec Law2Vec is a Word2Vec [40] variation, trained with 123,066 legal documents including legislation, court decisions, and statues. Word2Vec attempts to place each word in a vector space such that words with similar meanings are close together by iterating over the entire corpus. Each step considers a word as well as its surroundings within a window. It is observed that Law2Vec exhibits comparable performance to the other techniques in the task of similarity measuring between legal documents [36]. There are two sets of vectors in Law2Vec: one with 100 dimensions and one with 200 dimensions. The experiments are conducted using the 200-dimensional model.

SciBERT SciBERT is a BERT [18] model trained with the full text of 1.14M papers from the computer science and biomedical domain. BERT is a Transformer-based model trained for the tasks of language modeling and next-sentence prediction. Being pre-trained on a vast corpus of scientific literature, including papers from a range of fields like computer science, chemistry, and biology, SciBERT is particularly well-suited for various scientific domain tasks like text classification, named entity recognition, and question answering. SciBERT has two versions: cased and uncased. We utilized the uncased version during the experiments.

SPECTER SPECTER is a Transformer-based method to generate document embeddings, specifically for scientific texts. According to the authors, it differs from other pre-trained models with its applicability to downstream tasks such as CR without any task-specific fine-tuning.

LegalBERT Another applied BERT variant is called LegalBERT which is a model trained with various kinds of legal documents such as legislation, court cases, and contracts. Among the other variations, the uncased base model of LegalBERT is used for experiments.

SciNCL Another BERT-based model is SciNCL, which is initialized with SciBERT weights and uses controlled sampling during training. Authors claim that SciNCL outperforms SPECTER in various metrics including CR.

SGPT SGPT is a GPT-based [50] model, trained to obtain sentence embeddings for the task of semantic search. GPT is a powerful model for the task of next token prediction since it is trained for the purpose of language modeling. It leverages decoders to generate sentence embeddings that can be used for semantic search tasks. Through this approach, SGPT achieves superior performance compared to previous state-of-the-art sentence embedding methods.

4 Experiments & Discussions

4.1 Data Set & Experimental Setup

Articles used in this study are gathered from LawArXiv[7], which is an open-source scholarly legal article repository. It contains 1366 articles that cover 27 different legal subjects. A tool[8] for scraping search engine results page (SERP) is used to gather citing articles from Google Scholar[9] and more than 10K articles that cite LawArXiv articles are obtained. Then, the contents of PDF files are extracted using a Python package called pdfplumber[10]. Faulty extracted or non-English articles are removed after this step. Preprocessing extracted content involves converting whole text into lowercase and removing non-ASCII characters. Abstracts of the articles are obtained by splitting the document using the keyword "abstract". In the end, 719 LawArXiv articles with 10,111 citation links from 8,887 citing articles are used in the experiments.

In line with other content-based global citation recommendation studies [39], abstracts of the articles are used as input for the fine-tuning and inference steps for all approaches. We partitioned the dataset into separate training and testing sets, utilizing 70% of the data for training and 30% for testing purposes. Models are fine-tuned for three epochs and the triplet loss function is used. The obtained document and query embeddings from the pre-trained and fine-tuned models are used to calculate cosine similarity between query-document pairs, which is used to rank documents. Sentence-Transformers [51], a framework based on Huggingface's Transformers library [61], is used to make use of pre-trained models and fine-tuning.

[7] https://osf.io/preprints/lawarxiv
[8] https://serpapi.com/google-scholar-api
[9] https://scholar.google.com/
[10] https://github.com/jsvine/pdfplumber

4.2 Evaluation Metrics

Experiments' results are reported with three different metrics, which are *Mean Average Precision (MAP)*, *Recall*, and *Mean Reciprocal Rank (MRR)*. We chose to use n=10 as the reference number for computing metrics, as on average there are 14 citation links extracted per article.

MAP For a number (N) of queries, mean average precision is the mean of the average precision (AP) scores of each query (Q) as in the following formula:

$$MAP = \frac{1}{N} \sum_{i=1}^{N} AP(Q_i)$$

Recall The ratio of retrieved relevant documents (*True Positives*) to the total number of relevant documents (*True Positives + False Negatives*) is called recall. It is calculated with the following formula:

$$Recall = \frac{TP}{TP + FN}$$

MRR For a number (N) of queries, mean reciprocal rank is the mean of the reciprocal ranks where $rank_Q$ points the position of the first relevant document that is retrieved for a query (Q) as in the following formula:

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{rank_{Q_i}}$$

4.3 Results & Discussions

This section provides and discusses the results of our experiments under four subsections.

Table 1. Performance of BM25 and pre-trained models for retrieving top-10 articles.

	MAP@10 ↓	**Recall@10 ↓**	**MRR@10 ↓**
BM25	**0.26**	**0.43**	**0.31**
SciNCL	0.18	0.33	0.23
SGPT	0.17	0.30	0.22
SPECTER	0.14	0.26	0.19
Law2Vec	0.11	0.21	0.17
LegalBERT	0.08	0.16	0.15
SciBERT	0.08	0.16	0.14

Table 2. Performance of fine-tuned models for retrieving top-10 articles.

Fine-tuned	**MAP@10 ↓**	**Recall@10 ↓**	**MRR@10 ↓**
SciNCL	**0.26**	**0.49**	**0.30**
SciBERT	**0.26**	**0.49**	0.29
SPECTER	0.25	0.47	0.28
SGPT	0.25	0.46	0.29
LegalBERT	0.24	0.47	0.28

Table 3. Performance of pre-trained models for re-ranking top-10 articles retrieved by BM25.

BM25 Prefetch	**MAP@10 ↓**	**Recall@10 ↓**	**MRR@10 ↓**
SciNCL	**0.25**	**0.43**	**0.30**
SGPT	0.24	**0.43**	0.29
SPECTER	0.23	**0.43**	0.28
LegalBERT	0.19	**0.43**	0.24
Law2Vec	0.19	**0.43**	0.24
SciBERT	0.18	**0.43**	0.23

Table 4. Performance of fine-tuned models for re-ranking top-10 articles retrieved by BM25.

BM25 Prefetch + Fine-tuning	**MAP@10 ↓**	**Recall@10 ↓**	**MRR@10 ↓**
SciNCL	**0.29**	**0.43**	**0.34**
SGPT	**0.29**	**0.43**	**0.34**
SPECTER	0.28	**0.43**	0.33
SciBERT	0.28	**0.43**	0.33
LegalBERT	0.27	**0.43**	0.32

BM25 and Pre-trained Models: Comparison of BM25 and pre-trained models (Table 1) shows that SciBERT, which is trained with scientific documents, has no understanding of legal texts. On the other hand, the poor performance of Law2Vec and LegalBERT, which are language models specific to the legal domain, might be explained as they were not trained for the task of CR. Observable performance increase (from 0.08 to 0.24 MAP@10) of LegalBERT after fine-tuning for the CR task (Table 2) also supports this claim. The pre-trained SGPT (Table 1) outperforms other pre-trained models, except SciNCL, even though it is not trained on scientific texts. This occurs possibly because it is directly trained for semantic search which is a task highly related to CR. Hence, it is more successful at retrieving the top-k articles for a given query. Yet, BM25 exceeds the performance of all pre-trained models and shows that it is a strong baseline (0.26 MAP@10) for the legal CR task, performing on par with existing performances in the literature [39] for other domains.

Fine-tuned Models: When the pre-trained language models are fine-tuned (Table 2) using the gathered dataset for the task of CR, SciNCL and SciBERT performs relatively better than others, in accordance with claims of [46]. Performance increases on all models show that when domain knowledge is provided, those models are also able to adapt to the legal domain. Besides, the significant improvement in LegalBERT's performance suggests that the model can effectively leverage domain-specific knowledge once it is trained on the task at hand.

Pre-fetching with BM25 and Re-ranking with Pre-trained Models: Parallel with the results of the first experiment, SciNCL (0.25 MAP@10) stands out as the best-performing pre-trained model in re-ranking BM25's pre-fetched articles (Table 3). While the performance of all models is increased, they still do not reach the level of our baseline (BM25).

Pre-fetching with BM25 and Re-ranking with Fine-tuned Models: Table 4 shows the performance of fine-tuned models on re-ranking BM25's pre-fetched articles. All fine-tuned models increase the performance of BM25, and demonstrate greater success than the second experiment where there is no pre-fetching step. Overall, SciNCL and SGPT stand out as the best performing models among others, improving performance of BM25 (from 0.26 to 0.29 MAP@10).

5 Conclusion

Our study presents the first scholarly legal citation recommendation dataset in the literature, consisting of 719 scholarly legal articles with 10K incoming citations from 9K articles, to make up for the lack of scholarly legal articles in the scientific text datasets. Using the gathered dataset, experimental results with state-of-the-art models are reported in four different setups. In conclusion, our findings indicate that BM25 serves as a strong baseline for citation recommendation on scholarly legal articles, while combination of BM25 and SciNCL or SGPT

for pre-fetching and re-ranking, respectively, produces the most effective results on our dataset, increasing the performance of the baseline (BM25) from 0.26 to 0.29 MAP@10. It is clear that pre-trained language models perform well on this task when further trained with a suitable dataset, on which future research should focus. The size of the gathered dataset is small when compared to the other domains used in the citation recommendation literature. This might affect the performance of the models in learning the task and domain, even though the entire LawArXive collection is gathered, as stated in Section 4. As a future work, one may think of enlarging this dataset with articles from law journals not indexed within LawArXiv.

Acknowledgement

The numerical computations reported in this article are performed via TUBITAK ULAKBIM High Performance and Grid Computing Center.

References

1. Ali, Z., Qi, G., Kefalas, P., Abro, W.A., Ali, B.: A graph-based taxonomy of citation recommendation models. Artificial Intelligence Review **53**(7), 5217–5260 (Feb 2020). https://doi.org/10.1007/s10462-020-09819-4
2. Ali, Z., Ullah, I., Khan, A., Jan, A.U., Muhammad, K.: An overview and evaluation of citation recommendation models. Scientometrics **126**(5), 4083–4119 (Mar 2021). https://doi.org/10.1007/s11192-021-03909-y
3. Amami, M., Pasi, G., Stella, F., Faiz, R.: An LDA-based approach to scientific paper recommendation. In: Natural Language Processing and Information Systems, pp. 200–210. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-41754-7_17
4. Baez, M., Mirylenka, D., Parra, C.: Understanding and supporting search for scholarly knowledge. Proceeding of the 7th European Computer Science Summit pp. 1–8 (2011)
5. Beel, J., Gipp, B., Langer, S., Breitinger, C.: Research-paper recommender systems: a literature survey. International Journal on Digital Libraries **17**(4), 305–338 (Jul 2015). https://doi.org/10.1007/s00799-015-0156-0
6. Beltagy, I., Lo, K., Cohan, A.: Scibert: Pretrained language model for scientific text. In: EMNLP (2019)
7. Bhagavatula, C., Feldman, S., Power, R., Ammar, W.: Content-based citation recommendation. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/n18-1022
8. Bhattacharya, P., Ghosh, K., Pal, A., Ghosh, S.: Methods for computing legal document similarity: A comparative study. ArXiv **abs/2004.12307** (2020)
9. Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.Y., Lee, D., Powley, B., Radev, D., Tan, Y.F.: The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In: Proceedings of the Sixth

International Conference on Language Resources and Evaluation (LREC'08). European Language Resources Association (ELRA), Marrakech, Morocco (May 2008), http://www.lrec-conf.org/proceedings/lrec2008/pdf/445_paper.pdf
10. Bornmann, L., Haunschild, R., Mutz, R.: Growth rates of modern science: A latent piecewise growth curve approach to model publication numbers from established and new literature databases (2020). https://doi.org/10.48550/ARXIV.2012.07675
11. Brown, D., Sarthak Jain, Novotný, V., Nlp4whp: dorianbrown/rank_bm25: (2022). https://doi.org/10.5281/ZENODO.6106156
12. Brown, P., Kulkarni, A.S., Refai, O., Zhou, Y.: Large expert-curated database for benchmarking document similarity detection in biomedical literature search. Database: The Journal of Biological Databases and Curation **2019** (2019)
13. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: The muppets straight out of law school. In: Findings of the Association for Computational Linguistics: EMNLP 2020. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.261
14. Chalkidis, I., Kampas, D.: Deep learning in law: early adaptation and legal word embeddings trained on large corpora. Artificial Intelligence and Law **27**(2), 171–198 (Dec 2018). https://doi.org/10.1007/s10506-018-9238-9
15. Cohan, A., Feldman, S., Beltagy, I., Downey, D., Weld, D.: SPECTER: Document-level representation learning using citation-informed transformers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.207
16. Cui, J., Shen, X., Nie, F., Wang, Z., Wang, J., Chen, Y.: A survey on legal judgment prediction: Datasets, metrics, models and challenges (2022). https://doi.org/10.48550/ARXIV.2204.04859
17. Dadgostari, F., Guim, M., Beling, P.A., Livermore, M.A., Rockmore, D.N.: Modeling law search as prediction. Artificial Intelligence and Law **29**(1), 3–34 (Feb 2020). https://doi.org/10.1007/s10506-020-09261-5
18. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding (2018). https://doi.org/10.48550/ARXIV.1810.04805
19. Dhani, J.S., Bhatt, R., Ganesan, B., Sirohi, P., Bhatnagar, V.: Similar cases recommendation using legal knowledge graphs (2021). https://doi.org/10.48550/ARXIV.2107.04771
20. Färber, M., Jatowt, A.: Citation recommendation: approaches and datasets. International Journal on Digital Libraries **21**(4), 375–405 (Aug 2020). https://doi.org/10.1007/s00799-020-00288-2
21. Färber, M., Thiemann, A., Jatowt, A.: A high-quality gold standard for citation-based tasks. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (May 2018), https://aclanthology.org/L18-1296
22. Fetahu, B., Markert, K., Anand, A.: Automated news suggestions for populating wikipedia entity pages. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM (Oct 2015). https://doi.org/10.1145/2806416.2806531
23. Galke, L., Mai, F., Vagliano, I., Scherp, A.: Multi-modal adversarial autoencoders for recommendations of citations and subject labels. In: Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization. ACM (Jul 2018). https://doi.org/10.1145/3209219.3209236

24. Huang, Z., Low, C., Teng, M., Zhang, H., Ho, D.E., Krass, M.S., Grabmair, M.: Context-aware legal citation recommendation using deep learning. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. ACM (Jun 2021). https://doi.org/10.1145/3462757.3466066
25. Khadka, A., Knoth, P.: Using citation-context to reduce topic drifting on pure citation-based recommendation. In: Proceedings of the 12th ACM Conference on Recommender Systems. ACM (Sep 2018). https://doi.org/10.1145/3240323.3240379
26. Lao, N., Cohen, W.W.: Relational retrieval using a combination of path-constrained random walks. Machine Learning **81**(1), 53–67 (Jul 2010). https://doi.org/10.1007/s10994-010-5205-8
27. Li, X., Chen, Y., Pettit, B., Rijke, M.D.: Personalised reranking of paper recommendations using paper content and user behavior. ACM Transactions on Information Systems **37**(3), 1–23 (Mar 2019). https://doi.org/10.1145/3312528
28. Liang, Y., Li, Q., Qian, T.: Finding relevant papers based on citation relations. In: Web-Age Information Management, pp. 403–414. Springer Berlin Heidelberg (2011). https://doi.org/10.1007/978-3-642-23535-1_35
29. Liu, H., Kong, X., Bai, X., Wang, W., Bekele, T.M., Xia, F.: Context-based collaborative filtering for citation recommendation. IEEE Access **3**, 1695–1703 (2015). https://doi.org/10.1109/access.2015.2481320
30. Lo, K., Wang, L.L., Neumann, M., Kinney, R., Weld, D.: S2ORC: The semantic scholar open research corpus. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.447
31. Locke, D., Zuccon, G.: Case law retrieval: problems, methods, challenges and evaluations in the last 20 years (2022). https://doi.org/10.48550/ARXIV.2202.07209
32. Ma, S., Zhang, C., Liu, X.: A review of citation recommendation: from textual content to enriched context. Scientometrics **122**(3), 1445–1472 (Jan 2020). https://doi.org/10.1007/s11192-019-03336-0
33. Ma, X., Wang, R.: Personalized scientific paper recommendation based on heterogeneous graph representation. IEEE Access **7**, 79887–79894 (2019). https://doi.org/10.1109/access.2019.2923293
34. Ma, X., Zhang, Y., Zeng, J.: Newly published scientific papers recommendation in heterogeneous information networks. Mobile Networks and Applications **24**(1), 69–79 (Sep 2018). https://doi.org/10.1007/s11036-018-1133-9
35. Mahdabi, P., Crestani, F.: Query-driven mining of citation networks for patent citation retrieval and recommendation. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM (Nov 2014). https://doi.org/10.1145/2661829.2661899
36. Mandal, A., Ghosh, K., Ghosh, S., Mandal, S.: Unsupervised approaches for measuring textual similarity between legal court case reports. Artificial Intelligence and Law (Jan 2021). https://doi.org/10.1007/s10506-020-09280-2
37. McNee, S.M., Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S.K., Rashid, A.M., Konstan, J.A., Riedl, J.: On the recommending of citations for research papers. In: Proceedings of the 2002 ACM conference on Computer supported cooperative work. ACM (Nov 2002). https://doi.org/10.1145/587078.587096
38. Medic, Z., Snajder, J.: A survey of citation recommendation tasks and methods. Journal of Computing and Information Technology **28**(3), 183–205 (Jul 2021). https://doi.org/10.20532/cit.2020.1005160

39. Medić, Z., Šnajder, J.: Large-scale evaluation of transformer-based article encoders on the task of citation recommendation (2022). https://doi.org/10.48550/ARXIV.2209.05452
40. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013). https://doi.org/10.48550/ARXIV.1301.3781
41. Mu, D., Guo, L., Cai, X., Hao, F.: Query-focused personalized citation recommendation with mutually reinforced ranking. IEEE Access **6**, 3107–3119 (2018). https://doi.org/10.1109/access.2017.2787179
42. Muennighoff, N.: SGPT: GPT sentence embeddings for semantic search (2022). https://doi.org/10.48550/ARXIV.2202.08904
43. Nallapati, R.M., Ahmed, A., Xing, E.P., Cohen, W.W.: Joint latent topic models for text and citations. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM (Aug 2008). https://doi.org/10.1145/1401890.1401957
44. Nogueira, R., Cho, K.: Passage re-ranking with BERT (2019). https://doi.org/10.48550/ARXIV.1901.04085
45. Ostendorff, M., Ash, E., Ruas, T., Gipp, B., Moreno-Schneider, J., Rehm, G.: Evaluating document representations for content-based legal literature recommendations. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. ACM (Jun 2021). https://doi.org/10.1145/3462757.3466073
46. Ostendorff, M., Rethmeier, N., Augenstein, I., Gipp, B., Rehm, G.: Neighborhood contrastive learning for scientific document representations with citation embeddings (2022). https://doi.org/10.48550/ARXIV.2202.06671
47. Peng, H., Liu, J., Lin, C.Y.: News citation recommendation with implicit and explicit semantics. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/p16-1037
48. Pennock, D.M., Horvitz, E.J., Lawrence, S., Giles, C.L.: Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach (2013). https://doi.org/10.48550/ARXIV.1301.3885
49. Radev, D.R., Muthukrishnan, P., Qazvinian, V.: The ACL Anthology network. In: Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries (NLPIR4DL). pp. 54–61. Association for Computational Linguistics, Suntec City, Singapore (Aug 2009), https://aclanthology.org/W09-3607
50. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
51. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (11 2019), https://arxiv.org/abs/1908.10084
52. Ren, X., Liu, J., Yu, X., Khandelwal, U., Gu, Q., Wang, L., Han, J.: ClusCite. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM (Aug 2014). https://doi.org/10.1145/2623330.2623630
53. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at trec-3. In: Overview of the Third Text REtrieval Conference (TREC-3). pp. 109–126. Gaithersburg, MD: NIST (January 1995)
54. Rosa, G.M., Rodrigues, R.C., Lotufo, R., Nogueira, R.: Yes, bm25 is a strong baseline for legal case retrieval (2021). https://doi.org/10.48550/ARXIV.2105.05686

55. Saier, T., Färber, M.: Bibliometric-enhanced arxiv: A data set for paper-based and citation-based tasks. In: Cabanac, G., Frommholz, I., Mayr, P. (eds.) Proceedings of the 8th International Workshop on Bibliometric-enhanced Information Retrieval (BIR 2019) co-located with the 41st European Conference on Information Retrieval (ECIR 2019), Cologne, Germany, April 14, 2019. CEUR Workshop Proceedings, vol. 2345, pp. 14–26. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2345/paper2.pdf
56. Sugiyama, K., Kan, M.Y.: Scholarly paper recommendation datasets (2013). https://doi.org/10.25540/BBCH-QTT8
57. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: ArnetMiner. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM (Aug 2008). https://doi.org/10.1145/1401890.1402008
58. Thomas, M., Vacek, T., Shuai, X., Liao, W., Sanchez, G., Sethia, P., Teo, D., Madan, K., Custis, T.: Quick check: A legal research recommendation system. In: NLLP@ KDD. pp. 57–60 (2020)
59. Wahle, J.P., Ruas, T., Mohammad, S.M., Gipp, B.: D3: A massive dataset of scholarly metadata for analyzing the state of computer science research (2022). https://doi.org/10.48550/ARXIV.2204.13384
60. Winkels, R., Boer, A., Vredebregt, B., Someren, A.V.: Towards a legal recommender system. In: International Conference on Legal Knowledge and Information Systems (12 2014). https://doi.org/10.3233/978-1-61499-468-8-169
61. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T.L., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M.: Huggingface's transformers: State-of-the-art natural language processing (2019). https://doi.org/10.48550/ARXIV.1910.03771
62. Yang, L., Zhang, Z., Cai, X., Guo, L.: Citation recommendation as edge prediction in heterogeneous bibliographic network: A network representation approach. IEEE Access **7**, 23232–23239 (2019). https://doi.org/10.1109/access.2019.2899907
63. Yu, X., Gu, Q., Zhou, M., Han, J.: Citation prediction in heterogeneous bibliographic networks. In: Proceedings of the 2012 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics (Apr 2012). https://doi.org/10.1137/1.9781611972825.96

Programming Contract Amending

Cosimo Laneve[1], Alessandro Parenti[2], and Giovanni Sartor[2]

[1] Department of Computer Science and Engineering, University of Bologna, Italy
[2] Department of Legal Studies, University of Bologna, Italy

Abstract. Parties to a contract are generally free to modify it at their will, provided that they find an agreement. Moreover, real-world events can affect the contractual relationship independently from parties' intent and require an adaptation of the contract to the new circumstances. When the agreement is defined using a programming language, amendments likely entail runtime modifications to the contract code. In this paper, we analyse *higher-order Stipula*, an extension to the *Stipula* language to express and implement contract amendments. We believe that its high level of abstraction can help legal professionals reason about contract modifications in a simple and intelligible way. To test the new feature we provide two real-world examples of contract modifications and model them in *higher-order Stipula*.

1 Introduction

The use of computer code to represent, monitor or execute a legal agreement between parties has been researched and employed in some form since the 1970s [22]. The computerization of contracts can bring different benefits: lower costs of digital transactions, the monitoring of business procedures, or the avoidance of litigation because of an *ex-ante* automatic assessment of compliance.[23].

Stipula [7] is a domain-specific language developed for drafting legal contracts in computable form. It was designed with the principle of having an abstraction level as close as possible to legal contracts to facilitate its use to legal professionals. For this reason, it is based on a small set of primitives that reflect the basic elements of traditional contracts (*permissions, obligations* etc.).

In the present work we discuss the addition of a new feature to *Stipula*, the ability to provide for future amendments during contract execution. Indeed, there may be several reasons to modify an agreement. An amendment may be needed simply because parties change their mind, or because of an unexpected event that affects the contractual relationship. The latter cases are usually dealt with in contracts by *hardship* clauses, and represent a crucial issue especially in long-term commercial agreements. In Section 3, we analyse the most common scenarios requiring an amendment, and discuss the legal basis to it in different legal systems.

We believe that being able to model contract amendments is a necessary feature to build a powerful and complete language for legal contracts. Up to our knowledge, no other language has addressed this issue.

Current *Stipula* contracts are immutable, i.e., cannot be modified once the execution has been started. As a consequence, to model contract amendments, one would have to anticipate the potential modification cases at the time of contract formation, and implement them in code accordingly. Besides being hardly feasible, such a practice would significantly raise drafting costs, thus nullifying one of the main purposes for digitalizing legal contracts.

To address these issues, we present *higher-order Stipula*, an extension of the language that implements a high-level mechanism to manage runtime modifications of contract code. Through *higher-order*, we admit that function invocations may carry code as an input parameter that patches the contract protocol. This solution allows us to manage both situations where the modification affects the whole body of the contract code, replacing it with new one, as well as situations where only some parts are amended leaving the rest operational.

In Section 4, we test the new feature on two real-world examples directly taken from contract practice. We end our contribution by discussing the state of the art in Section 5 and presenting our conclusions in Section 6.

2 Background: modelling contracts with *Stipula*

Stipula is a domain-specific language for modelling legal contracts that has been designed to be an intermediate programming language, more concrete and execution-oriented than a specification language and, at the same time, more abstract than a full-fledged programming language [7]. *Stipula* is designed on top of a small set of primitives that reflect some of the main features of legal contracts:

- a contract enters into force at the moment of parties' agreement, so called 'meeting of the minds'. This is represented by the *agreement* constructor through which parties are called to agree to the terms of the contract;

```
agreement (Buyer, Seller) {
    Buyer, Seller : cost, time_due
} init ⇛ @Inactive
```

- legal contracts may create, extinguish or regulate normative positions linked to parties such as permissions, prohibitions, obligations or powers. *States* are used to model and automatically enforce prohibitions and permissions by precluding or allowing the invocation of functions. They are indicated by an "@" in front.

```
@Inactive
```

  Events are used to check the fulfilment of obligations at a certain time and eventually issuing a penalty;

```
now + time_due >> @Inactive{
    Termination → Buyer
}⇛ @End
```

– contracts are usually required to manage currency or digital goods (transfers, escrows etc.). In *Stipula* these entities are called ***assets*** and, in order to model and manage them, an *ad hoc* syntax is employed to mark a separation from other data types or variables. An asset transfer is expressed as follows;

```
1    t —∘ token
```

– in contrast to assets, contract terms such as cost or deadlines are represented by `fields` of the *Stipula* contract and may be set and agreed to during the agreement phase. A field update is expressed as follows;

```
1    x → cost
```

– in most cases, some aspects of contracts' execution depend on the occurrence of external events whose related data need to be fed into the contract. This need is implemented through an ***Authority***, a trusted intermediary which participates to the contract just like a party and which is the only allowed to call specific functions. The same solution is used to implement dispute resolution and the verification of non-automatically verifiable (also ambiguous) circumstances (e.g. *force majeure*, serious damage, etc.);

As an example, consider the *Stipula* contract in Figure 1 regulating a bike rental service. Lines 1-3 define the name of the contract and the list of *assets* and *fields* that are used therein; the code in lines 5-7 is meeting a `Lender` and a `Borrower` to agree on both the `rentingTime` and its `cost`. After the agreement, the contract starts and goes into the state `@Inactive` expressing that no rent will occur until the payment.

When the contract is in state `@Inactive`, the `Lender` can invoke the `offer` function to make available to the `Borrower` an access code necessary to unlock the bike. This is stored in the `code` field. Once the code has been received, the contract moves to the state `@Payment` (line 10) that allows the `Borrower` to pay and activate the rental.

The function `pay` in lines 12-18 is defining the payment of the rental by `Borrower`, which sends an asset `h` – asset arguments are indicated in square brackets – to the contract. The function call has a precondition – operation `h == cost` – that checks whether the borrower pays the correct fee or not. The semantics of the operation `h —∘ wallet` in line 15 (an abbreviation for `h —∘ h, wallet`) is that, after the execution, `h` is not owned by `Borrower` anymore and is taken by the contract that stores it in the *asset field* `wallet`.

Once the fee has been payed, the `Borrower` gets the access code and the contract transits into a `@Using` state. Lines 15-18 illustrate how *obligations* are specified in *Stipula* by means of *events*. That is, the function `pay` issues a commitment that is checked at a specific time limit: if the `rentingTime` (specified in the agreement) is reached and the bike has not been returned yet (the state of the contract is still `@Using`) then a message of returning the bike is sent to the `Borrower` (`"End_Reached"` → `Borrower` in line 17) and the contract moves to the state `@Return`. We remark that events are not triggered by any party: they are automatically executed when the time condition is met.

```
1  stipula Bike_Rental {
2      assets wallet
3      fields cost, rentingTime, code
4
5     agreement (Lender,Borrower,Authority){
6          Lender, Borrower: rentingTime, cost
7        } ⇒ @Inactive
8
9     @Inactive Lender : offer(x) {
10         x → code     } ⇒ @Payment
11
12    @Payment Borrower : pay[h] (h == cost) {
13         h ⊸ wallet
14         code → Borrower
15         now + rentingTime ≫
16            @Using {
17               "End_Reached" → Borrower
18                } ⇒ @Return    } ⇒ @Using
19
20    @Using Borrower : end {
21         now → Lender      } ⇒ @Return
22
23    @Return Lender : rentalOk {
24         wallet ⊸ Lender      } ⇒ @End
25
26    @Using,@Return Lender,Borrower : dispute(x) {
27         x → _     } ⇒ @Dispute
28
29    @Dispute Authority : verdict(x,y)
30       (y>=0 && y<=1) {
31          x → Lender   x → Borrower
32          wallet×y ⊸ wallet, Lender
33          wallet ⊸ Borrower       } ⇒ @End
34 }
```

Fig. 1. The Bike Rent contract

The termination of the rental further requires the `Lender` to confirm the absence of damages before receiving the fee (function `rentalOK` in lines 23-24). For the sake of simplicity this contract does not impose a penalty to the `Borrower` for late return, but it is not difficult to modify the rental contract by requiring the borrower to pay a higher fee that is deposited as security and is reimbursed in case of timely return [8].

Lines 27-34 illustrate disputes resolution in *Stipula*, which somehow mimic the behaviour of a court. In fact, when contract's violations cannot be checked by the software, such as the damage or misuse of the bike, it is necessary the involvement of a trusted third party, the `Authority` (which must have been included in the agreement), to supervise the dispute and to provide a trusted resolution mechanism. More concretely, the `Authority` takes the legal responsibility of interfacing with a court or an Online Dispute Resolutions platform[3]. The function `dispute` may be invoked either by the `Lender` or by the `Borrower`, either in the state `@Using` or `@Return`, and carries the reasons for kicking the dispute off (`x` is intended to be a string). Once the reasons are communicated to every party (we use the abbreviation "`_`" instead of writing three times the sending operation) the contract transits into a state `@Dispute` where the `Authority` will analyze the is-

[3] as The European ODR platform at `https://ec.europa.eu/consumers/odr`.

sue and emit a verdict. This is performed in `@Dispute` where only the invocation of the `verdict` function (line 30) is permitted. This function has two arguments: a string of motivations `x`, and a coefficient `y` that denotes the part of the wallet that will be delivered to `Lender` as reimbursement; the `Borrower` will get the remaining part. It is worth to spot this point: the statement `wallet×y ⊸ wallet, Lender` in line 33 *takes* the `y` part of `wallet` (`y` is in [0..1]) and sends it to `Lender`; *at the same time* the `wallet` is reduced correspondingly. The remaining part is sent to `Borrower` with the statement `wallet ⊸ Borrower` (which is actually a shortening for `wallet×1 ⊸ wallet, Borrower`) and the `wallet` is emptied.

Further transpositions of legal contracts in *Stipula* can be found in [6]. The basic definition of *Stipula* does not admit the management of *exceptional behaviours*, *i.e.* all those behaviours that cannot be anticipated due to the occurrence of unforeseeable and extraordinary events, which, in legal contracts, are usually dealt with amendments. The extension of the language with a feature for modelling amendments is discussed in the following sections.

3 Amending contracts

The principle of *freedom of contract* allows parties to modify the contract at their will, provided that there is an agreement and that its content is not against the law. Occasionally, one party may yield to the other the power to change some parts of the agreement unilaterally [3]. This is a common practice for consumer contracts and standard terms of service, where the right to modify is usually tied to certain requirements, such as notifying the other party of the change and giving them the possibility to withdraw. In specific cases, the right to unilateral modification (*jus variandi*) may be directly conferred by the legislator (e.g., in Italian law, the employer's right to change employee's tasks [4]).

The need to modify a contract may also derive from outside the parties, such as when a court declares a contract partially void due to formal or substantial flaws or when unexpected events outside the control of parties affect the contractual relationship. These last cases are particularly relevant in long-term contracts and require legal solutions in order to deal with occurrences that couldn't be anticipated in advance by the parties.

Contracts are entered into with the expectation that both parties will fulfil their obligations as agreed upon. The roman brocard *pacta sunt servanda* (agreement have to be respected), constitutes a foundational principle of contract theory: the contract is a mutual promise in which each party can hold the other one to the promised performance However, parties accepted to be bound by those promises under the particular set of circumstances standing at the time of stipulation: if these circustamces change, this commitments may need to be revised. For example, the beginning of a war could drastically raise the price of commodities needed for production or the outbreak of a pandemic could

[4] Art. 2103 Codice Civile

halt factories' activity. Such changes of circumstances may make performance of contractual obligations impossible, excessively onerous or even deprive it of its original utility for the counterparty. To address these situations, a legal basis to excuse performance or to legitimately request an amendment of the contract is provided by the principle of *clausola rebus sic stantibus*: a contract is binding only as far as the relevant circumstances remain the same as they were at the time of conclusion of the agreement [25].

The matter is known to most legal systems but it is addressed in different ways. In common law systems, courts elaborated the doctrine of *frustration*[5]. This represents an excuse when an unforseen change in circumstances deprives the contract of all the utility for one party, even though the material capacity to perform the obligation is not affected at all. In the United States, one can also find the notion of *impracticability* which, recognised by the Uniform Commercial Code (§ 2-615), offers a defense in case performance became impractical due to a contingency that parties excluded at the time of stipulation. Unlike under *frustration*, here performance may become extremely difficult or onerous for one party[20].

In civil law countries, the issue is often touched by national legislation. For example, in France and Italy, the respective civil codes provide a specific provision dealing with supervening events that render the performance excessively onerous for one party (Art. 1195 Code Civìl, art. 1467 Codice Civile). These norms give to the burdened party the possibility to request an amendment of the contract in order to recover the original contractual balance.

In contract practice, especially in international context, the eventuality of unexpected changes in circumstances that might affect the agreement is usually dealt with by providing specific clauses defining the conditions and procedures to be followed in such cases. Doing so, parties can avoid the uncertainty of being at the mercy of the relevant national legislation [11]. The main examples in this sense are *force majeure* and hardship clauses. While the former occurs when performance becomes impossible and usually leads to termination, the latter is activated when the equilibrium of a contract is altered making it significantly more onerous for one party[6]. In these cases, the burdened party is usually entitled to request an amendment of the contract, or its termination.

The contracts defined in the language of Section 2 are immutable. Therefore, in order to model either *force majeure* or hardship, one should anticipate all the appropriate amendments for each possible circumstance at the time of first drafting. While this is easy for termination clauses (it is enough to include a transition to a final state), it is clearly impossible for generic amendments [18]. Even an attempt to do that would raise drafting costs and introduce huge complexities in the contract, thus nullifying one of the main objectives of *Stipula*,

[5] The *frustration* doctrine was originally developed by English courts as a consequence to the famous *Coronation* cases in 1902-1904. The cancellation of King Edward VII's coronation frustrated the purpose of the defendants who leased apartments to witness the procession from a privileged spot. See *Krell v Henry* (1903).

[6] Art. 6.2.2 UNIDROIT Principles

which is to have a simple and intelligible code. For these reasons, in the next section, we discuss an extension of the language with a feature that allows parties to remove or amend the effects of a contract in a direct and intelligible way.

4 The *higher-order Stipula*

Technically, amendments are runtime adjustments to the contract's behavior. In programming languages, these runtime adjustment are usually expressed by *higher-order functions*, which are functions that may also take code as an input parameter. The input code goes into execution when the function is invoked, thus possibly modifying the former behaviour. A *higher-order* function in *Stipula* looks like this:

```
@Active Authority: disputeResolution (| X, Y, Z |) {remove X add Y run Z}
```

This function carries three parameters in brackets $(\!|\cdot|\!)$, the role of which is indicated by the directives `remove` X `add` Y `run` Z. X is a sequence of function names that will be removed from the contract (the terms are `f`, `A:f`, `Q A:f`); Y represents the new code added and may include declarations of new parties, fields, assets as well as new functions that will amend the contract. Notice that higher-order functions do not have a body: this is defined in Z in the form $\{..\} \Rightarrow$ `@Q`, and may also include the new elements defined in Y. It is worth mentioning that, while X and Y are potentially empty sequences (i.e., optional parameters), Z is mandatory (it is, in fact, necessary to at least define the state that the contract will transit to after the function invocation). A precise explanation of the syntax and formal semantics of *higher-order Stipula* is outside the scope of the present work. All the technical aspects are defined and presented in [16]. The purpose of this paper is instead to discuss the underlying legal basis to the new *higher-order* feature and to provide practical implementations of the language. We will do this by presenting two real-world cases of contract amendment.

The first one is taken from a dispute brought to the ICC International Court of Arbitration in 2001[7] and deals with price revision in long-term supply contracts, a recurrent situation in international commercial contracts [2]. The second one is taken from an Italian court case during the Covid pandemic which clearly produced a significant change in circumstances affecting many contractual relationships [24].

In the following examples, for purposes of conciseness, the *agreement* clause (presented in Section 2) is not included. This is substituted by the indication of the initial state (`Init`).

Example 1 The dispute of the first case concerned a long-term contract for the purchase of liquid natural gas. Parties decided to set the gas price by referring to a formula published by a price reporting agency, instead of referring to the market price. They also included a price revision clause requiring parties'

[7] ICC International Court of Arbitration case n. 10351/2001

agreement to proceed to the adjustment. The contract can be represented in *higher-order Stipula* as follows:

```
1   stipula GasSupply {
2       parties: Seller, Buyer, PriceProvider
3       fields: price, order
4       assets: wallet
5       Init: @Waiting_Order
6
7       @Waiting_order PriceProvider: update_price (x)[] {
8           x → price
9           x → Seller
10          x → Buyer } ⇛ @Waiting_order
11
12      @Waiting_order Buyer: place_order (x)[t] (x × price == t){
13          x → order
14          x → Seller
15          t ⊸ wallet } ⇛ @Order
16
17      @Order Seller: send_gas ()[g] (g == order){
18          g ⊸ Buyer
19          wallet ⊸ Seller } ⇛ @Waiting_order
20
21      @Waiting_order Seller,Buyer: price_revision(| X, Y, Z |){remove X add Y run Z}
22  }
```

Fig. 2. Gas Supply contract

A the beginning (lines 1-4), parties, fields, and assets are declared. The contract includes a third party, the `PriceProvider` which represents the reporting agency called to feed the updated price into the contract. The contract is initialised in the state `@Waiting_order` (line 4). Then, contract functions are declared. By calling `update_price`, the `PriceProvider` updates the gas price (line 8) and communicates it to both parties (lines 9-10). The `place_order` function can be used by the Buyer to make a purchase order of gas. This function takes in two parameters: `x` and `t`, respectively representing the amount of gas ordered and the currency. The parameter `x` is used to update the field `order` and is communicated to the seller; `t` is escrowed by the contract and stored in `wallet` (line 15). At this point, the contract transits to the `@Order` state thus enabling the seller to call `send_gas` function. Through this, the gas (`g`) is sent to the buyer and the seller automatically receives the money in `wallet`. The price revision clause is represented by the higher-order function `price_revision` that allows either `Seller` or `Buyer` to modify the contract code.

After some time, the referring agency modified the gas price formula, resulting in a relevant price increase compared to the market price [2]. The buyer opposed to the application of the new formula and requested the return to the previous one. Failing the attempt to find an agreement with the seller in this direction, the buyer resorted to arbitration which eventually supported the claim. For the purposes of our example, let's assume that parties found an agreement on price revision without going to litigation. The amendment would entail the substitution of the price provider with a new one and the disabling of the old function to update price that could be called by the old provider. To this end, we preclude the transition to `@Waiting_order` and introduce a whole new set of states. Therefore, `Buyer` invokes

```
1    Buyer: price_revision (|ε, 𝔻, {"new_PriceProvider" → Seller} ⇛ @New_Waiting_order|)
```

where ε is an empty sequence, indicating that there is no function to remove, and $\mathbb{D}$ is

```
1    parties: PriceProvider2
2
3        @New_Waiting_order PriceProvider2: update_price (x)[] {
4            x → price
5            x → Seller
6            x → Buyer
7        } ⇛ @New_Waiting_order
8
9        @New_Waiting_order Buyer: place_order (x)[t] (x × price == t){
10           x → order
11           x → Seller
12           t ⊸ wallet
13       } ⇛ @New_Order
14
15       @New_Order Seller: send_gas ()[g] (g == order){
16           g ⊸ Buyer
17           wallet ⊸ Seller
18       } ⇛ @New_Waiting_order
19
20       @New_Waiting_order Seller,Buyer: price_revision (|X, Y, Z|)
```

The third parameter of the `price_revision` function (the one replacing Z) defines its body. It communicates the identity of the new price provider to the `Seller` and makes the transition to the new state `@New_Waiting_order`. At this point, none of the functions that can be invoked from this state (or in a state reachable therefrom) provides for a transition to any of the previous states, thus completely disabling the old contract code. In fact, $\mathbb{D}$ defines a whole new set of states, functions, and introduces a new party to the contract (line 1) which is the only one entitled to call the new `update_price` (i.e., the new price provider).

In this case, the modification is implemented by precluding the invocation of old functions and not by removing them completely. This can be useful would parties want to go back to the original version of the contract at a later time. It will be sufficient to invoke the higher-order function `price_revision` also provided in the new version of the contract, and introduce a transition to one of the old states. This example represents a case of *additive amendment*, *i.e.* the modification affects the whole contract and the old functions are completely replaced by new ones.

Example 2 The second case study concerns the adaptation of a contract to new circumstances that is mandated by a court. While in certain countries, courts have historically been extremely rigorous in defending the sanctity of contracts, not allowing renegotiation or excuse for performance unless this became actually impossible (e.g., France)[17], other legal systems are less restrictive. This tendency is exemplified by some Italian case-law during the coronavirus pandemic, in which courts have imposed temporary modifications to contracts in force of the general principle of good faith in contractual relationship ex art. 1375 of Italian Civil Code [24]. Our example is taken from a claim discussed before the Rome's Tribunal in 2020[8].

[8] Tribunale Roma, ord. 27/08/2020

The case concerned a real estate rent contract for commercial use; the `Tenant` claimed a rent reduction because of the lack of revenues of his commercial activity during the lockdown months. In addition, the claimant sought to prevent the creditor from liquidating the surety provided for the non-performance. More precisely, the contract envisaged a rent fee of 8000 euros (`monthly_rent`) to be paid on the 30th of every month (`t_due`); it also included a surety in case of non-performance by the `Tenant`, to be provided by a bank (`Guarantor`). The contract can be defined in *higher-order Stipula* as follows:

```
1   stipula CommercialRent {
2       parties: Lessor, Tenant, Guarantor, Authority
3       fields: monthly_rent, t_due, n_month, debt
4       Init: @Inactive
5
6       @Inactive Lessor: delivery ()[k] {
7           k ⊸ Tenant                          #k = property key
8           0 → n_month
9           now + t_due >> @Running (n_month == 0){
10              "Rent_Due" → Tenant
11              "Rent_Due" → Guarantor
12              monthly_rent → debt
13          } ⇛ @Delay                  } ⇛ @Running
14
15      @Running Tenant: pay (n)[t] (t == monthly_rent && n == n_month){
16          t ⊸ Lessor
17          n_month + 1 → n_month
18          now + t_due >> @Running (n_month == n + 1){
19              "Rent_Due" → Tenant
20              "Rent_Due" → Guarantor
21              debt + monthly_rent → debt
22          } ⇛ @Delay                  } ⇛ @Running
23
24      @Delay Lessor: continue (n)[] (n == n_month){
25          "continue_with_debt" → Guarantor
26          now + t_due >> @Running (n_month == n){
27              "Rent_Due" → Tenant
28              "Rent_Due" → Guarantor
29              debt + monthly_rent → debt
30          }⇛ @Delay                   } ⇛ @Running
31
32      @Delay Lessor: claim_liquidation ()[] (debt ≠ 0){
33          "request_liquidation" → Guarantor     } ⇛ @Delay
34
35      @Delay Guarantor: liquidate ()[t] (t == debt){
36          t ⊸ Lessor                  } ⇛ @Running
37
38      @~ Authority: dispute_resolution ⟪X, Y, Z⟫ {remove X add Y run Z}
39  }
```

Fig. 3. Real Estate Commercial Rent

Beside the `Guarantor`, the contract also includes the `Authority` as a party (in our case, the Tribunal) that can intervene on the contract in case disputes arise. This is can be done by calling the `dispute_resolution` function (the $\sim$ indicates that the function can be called in any state).

At the beginning, the `Lessor` can use the `delivery` function to send the key of the rented property to the `Tenant` (in this case, the key is represented by a token allowing the access to the building) and formally start the rent. Within the due time (i.e., 30th of every moth), the `Tenant` can pay the rent by calling `pay` and sending the correct amount of currency to the `Lessor` (line 15).

The obligation to pay and the eventual surety liquidation are modelled through an event (lines 9-13 and 18-22). In case the monthly instalment is not payed within the due time (`t_due`), the `Tenant` and the `Guarantor` are notified with a warning (`"Rent_Due"`) and the contract transits to the state `@Delay`. This state allows the `Lessor` to claim a surety liquidation by the guarantor. As an alternative, the `Lessor` can also decide to go on without resorting to the `Guarantor` by calling the `continue` function. Nevertheless, the amount of unpaid rent is consistently updated and stored in the `debt` field (lines 12 and 21).

Because of the pandemic outbreak and the forced lockdown of commercial activities, the `Tenant` fails to perform the rent payment. When the `Lessor` requests the surety liquidation from the bank, the `Tenant` addresses the court to stop him and to adapt the contract to the changed circumstances. The judge eventually supports this claim and, in light of the principle of good faith in contractual relationships, provides for a reduction of the rent by forty percent and for the suspension of the the surety liquidation up to the threshold of thirty thousand euros. Both measures are temporary and, for the purposes of the present example, let's assume they are provided for six months. To implement the mandated amendments into the contract, the judge (`Authority`) invokes:

```
1 Authority : dispute_resolution(⟪ claim_liquidation, 𝔻, B ⟫)
```

where the new code, $\mathbb{D}$, is

```
1    fields: s_threshold, t_adapt
2
3    @Delay Lessor: claim_liquidation ()[] (debt ≥ s_threshold){
4            "request_liquidation" → Guarantor     } ⇏ @Delay
```

and the body, B, is

```
1    {  monthly_rent x (1 - 0.4) → monthly_rent
2       30 000 → s_threshold                    #30 000 euros
3       6m → t_adapt                            #6 months
4       now + t_adapt >> @~ {
5            monthly_rent / (1 - 0.4) → monthly_rent
6            0 → s_threshold} ⇏ @Running     } ⇏ @Running
```

In this case, the directive to execute is *remove* `claim_liquidation`, *add* $\mathbb{D}$, *run* B. This removes the function `claim_liquidation` from the old code and adds the new one in $\mathbb{D}$. $\mathbb{D}$ also declares two new fields `s_threshold` and `t_adapt`: the first one represents the new threshold amount of debt necessary to claim the surety liquidation; while the second is the duration of the temporary measures mandated by the judge (both fields are initialised in B). The new `claim_liquidation` in $\mathbb{D}$ (lines 3-5) has a new guard that precludes its invocation when the amount of debt is lower than the mandated threshold.

In the body B, `monthly_rent` is reduced by forty percent (line 1) and the new fields introduced in $\mathbb{D}$ are initialised (lines 2-3). Lines 4 to 6 define an event used to automatically remove the temporary measures at the end of the six months. The rent cost is brought back to the original level (line 5) and the threshold is removed by updating the field to zero (line 6).

In contrast with the previous example, here the modification affects only one part of the contract and is operated by completely deleting one function.

Doing so, we can keep both old and new codes operative and coexisting, while at the same time avoiding potential overlaps and conflict that can arise between functions with the same name. We believe that this possibility allows modelling contract amendments in a more simple and straightforward way.

5 Related Works

The digital representation of legal contracts has long been explored, for the main purpose of monitoring and automating contract-related procedures [19,12]. In this context, substantial work has been done in the wake of 'Ricardian Contracts', originally introduced by Ian Grigg in 1996 [14]. This approach consists in linking written contract documents with the related computer executable code via *parameters*. Through the use of mark-up languages, the natural language document is annotated to indicate which parts of the contract are the values to be inputted to the code. Further works extended this approach by building a template model for contracts [5] and providing specifications to increase contract's intelligibility [21]. However, the capability of capturing the semantics of an agreement by annotating natural language documents is limited to the input that is provided by the tagged data. Moreover, operational code may still remain opaque to legal professionals, thus preventing the validation of whether it is faithful to the actual agreement [4].

Declarative programming, having advantages in representing certain provisions and issues pertaining to efficient implementation, has also been explored for the modelling of legal contracts. An evaluation of its benefits and constraints, compared to imperative approaches, is given in [13].

A different approach to express contracts is represented by Domain-Specific languages (DSL). A well-designed, relatively understandable DSL for legal contracts has the advantage of keeping code and agreement (or a straightforward representation of it), within a single artefact. With a single artefact to deal with, it is simpler to check whether the meaning of the agreement and its code implementation match [4]. Such a contract can still be coupled with natural language explanations of the meaning of the code, but the code, rather than these explanations would provide the binding formulation of the contract. Different formalism and approaches have been studied in the literature. For instance Flood and Goodenough have described a loan agreement (in the financial domain) as a particular kind of finite state machine [10]. These machines are mathematical entities used to describe systems with finite set of states and transitions, where transitions allow movements from a state to another in response to given inputs (*events*). While this approach is interesting when the contract is simple enough, it becomes cryptic when the contract is more complex. In particular, it becomes hard to connect the machine to the standard formulation of the contract in natural languages.

Another interesting technique is based on Controlled Natural languages (CNL) [9,1]. A CNL resembles natural language in wording, but is based on formally defined syntax that is automatically converted to a programming language. As a

consequence, the code is easily readable. However, due to the constrains imposed by the CNL, it may result harder to write the contract (with respect to natural language) because the formalism may miss computational constructs. It has also been argued that a CNL might represent a "false friend" for the user [15], i.e., it might induce the user to assume that a CNL-expression has the same meaning as natural language expression, which might not always be the case.

The *higher-order Stipula* is a DSL that is based on state-oriented programming with explicit management of assets and with higher-order to express runtime modifications of the code. In our formalism, states are not finitely many because the contracts have memories that store settings and assets. Rather, states are used to express permissions and prohibition of invoking functionalities by contract parties.

6 Conclusions

The present work showcased an extension of *Stipula* for amending legal contracts at run-time. The extension relies on higher-order functions and allows one to program situations were the old code is completely replaced by new one as well as situations where old and new code are both operative and coexist. Overall, we believe that the higher-order mechanism is a simple and intelligible feature that may assist legal practitioners in programming contract amendments.

Up-to our knowledge, *higher-order Stipula* is the first legal contract language natively integrating amendments in its syntax. This has the advantage that amendments may be analysed using the same techniques on which the first-order language is based. The compliance of the types of the amendments with respect to the types of the original code is done by using the same original type inference system. By exploiting this property, for example, one can design techniques for constraining amendments using syntactic directives, as we do in [16]. Adding a runtime extension to some existing tool that copes with amendments is not the same as it would be unconstrained.

Future works shall also focus on the possible representation of parties' agreement within the contract in correspondence of runtime amendments, which is not dealt with here. This could be achieved by employing the agreement feature already used to represent the *meeting of the minds*.

We also remind that *higher-order Stipula* provides a user-friendly tool interface to help writing contract code as well as a prototype to execute and test code's behaviour. Both are available at [8].

References

1. Lexon language (2022), `http://lexon.org/`, (accessed: 13.04.2023)
2. Bortolotti, F., Ufot, D.: Hardship and force majeure in international commercial contracts: dealing with unforeseen events in a changing world. Kluwer Law International BV (2019)

3. Caldarelli, G.: Unilateral modification of long term contracts: American change of terms clauses and italian *Ius Variandi* from a 'relational' point of view. European Review of Contract Law **17**(1), 37–53 (2021)
4. Clack, C.D.: Languages for smart and computable contracts. arXiv preprint arXiv:2104.03764 (2021)
5. Clack, C.D., Bakshi, V.A., Braine, L.: Smart contract templates: foundations, design landscape and research directions. arXiv preprint arXiv:1608.00771 (2016)
6. Crafa, S., Laneve, C.: Programming legal contracts - A beginners guide to *Stipula*. In: The Logic of Software. A Tasting Menu of Formal Methods. Lecture Notes in Computer Science, vol. 13360, pp. 129–146. Springer (2022)
7. Crafa, S., Laneve, C., Sartor, G., Veschetti, A.: Pacta sunt servanda: legal contracts in *Stipula*. Science of Computer Programming **225**, 102911 (2023)
8. Crafa, S., Laneve, C., Veschetti, A.: The Stipula prototype, `https://github.com/stipula-language/stipula`, (accessed: 31.03.2023)
9. Datoo, A., Kowalski, R.: Logical English meets legal english for swaps and derivatives. Artificial Intelligence and Law (2021)
10. Flood, M.D., Goodenough, O.R.: Contract as automaton: representing a simple financial agreement in computational form. Artificial Intelligence and Law pp. 1–26 (2022)
11. Fontaine, M., De Ly, F.: Drafting international contracts. BRILL (2006)
12. Governatori, G.: Representing business contracts in RuleML. International Journal of Cooperative Information Systems **14**(02n03), 181–216 (2005)
13. Governatori, G., Idelberger, F., Milosevic, Z., Riveret, R., Sartor, G., Xu, X.: On legal contracts, imperative and declarative smart contracts, and blockchain systems. Artificial Intelligence and Law **26**, 377–409 (2018)
14. Grigg, I.: The Ricardian Contract (1996), `https://iang.org/papers/ricardian_contract.html`, (accessed: 13.04.2023)
15. Idelberger, F.: The uncanny valley of computable contracts: analysis of computable contract formalisms with a focus towards controlled natural languages. Ph.D. thesis, Europeen University Institute (2022)
16. Laneve, C., Parenti, A., Sartor, G.: Legal contracts amending with *Stipula* (2023), forthcoming publication at COORDINATION 2023
17. Lutzi, T.: Introducing imprévision into french contract law: A paradigm shift in comparative perspective. Ius Commune Europaeum, Intersentia (2016)
18. Mik, E.: Smart contracts: terminology, technical limitations and real world complexity. Law, innovation and technology **9**(2), 269–300 (2017)
19. Milosevic, Z., Gibson, S., Linington, P., Cole, J., Kulkarni, S.: On design and implementation of a contract monitoring facility. In: Proceedings. First IEEE International Workshop on Electronic Contracting, 2004. pp. 62–70 (2004)
20. Palmer, V.V.: Excused performances: Force majeure, impracticability, and frustration of contracts. The American Journal of Comparative Law **70**(Supplement_1), i70–i88 (2022)
21. Palmirani, M., Cervone, L., Vitali, F.: Intelligible Contracts. In: 53rd Hawaii International Conference on System Sciences. pp. 1780–1789 (2020)
22. Pfeiffer, H.K.: The diffusion of electronic data interchange. Springer Science and Business Media (2012)
23. Surden, H.: Computable Contracts. UCDL Rev. **46**, 629 (2012)
24. Torsello, M., Viglino, G.: Covid-19, hardship and force majeure before italian courts (March 2021), `https://tinyurl.com/45a5uesx`, (accessed: 13.04.2023)
25. Zimmermann, R.: The law of obligations: Roman foundations of the civilian tradition. Juta and Company Ltd (1990)

Understanding Privacy By Formalizing It*

Réka Markovich[1], Truls Pedersen[2], and Marija Slavkovik[2]

[1] University of Luxembourg reka.markovich@uni.lu
[2] University of Bergen
{Marija.Slavkovik,Truls.Pedersen}@uib.no

Abstract. In most of the modern societies, there is a broad consensus regarding the need for promoting privacy and thus placing restrictions on technological—including AI—developments to protect people's right to privacy. In order to meet these expectations on the algorithmic level, first we need to make the concept of privacy and the related or derived rights formally specified. However, the notion of (the right to) privacy is subject to different interpretations. In this paper, we use a multi-modal logic to provide an initial formalization of different theories and approaches' basic principles and their implications investigating the right to privacy as an epistemic right within the theory of normative positions.

Keywords: privacy · legal knowledge representation · normative positions.

1 Introduction

In the context of ethical impact of artificial intelligence, privacy is often discussed as a value eroded by digitization and artificial intelligence [Forbrukerrådet, 2018]. Privacy, however, is not one of the traditional moral values [Quine, 1978, Kinnier et al., 2000, Floridi and Cowls, 2019]. There are numerous attempts in the literature on defining privacy, but there is no consensus [Matzner and Ochs, 2019]. The overall privacy situation is made more confusing by what [Elvy, 2017] calls "emerging personal data economy". The data economy both exploits and drives the need for more specific privacy regulations.

On the global scale, privacy is a culturally divisive value or reference point, but in the so-called western countries there seems to be a broad consensus regarding its primary importance. This involves a vast regulative aspiration aiming at reasonable restrictions on the different technological developments. We believe that on the long run, the implementation of regulative expectations or imposing self-regulative restrictions, will require formal specification on what privacy can mean and what the right to privacy exactly is. If we take seriously the numerous claims that artificial intelligence in particular, and digitization in general, undermine the existences of privacy, then we need to have a good understanding of what privacy is, what duties the right to it implies, and how it can be preserved.

* This work was supported by the Fonds National de la Recherche Luxembourg through the project Deontic Logic for Epistemic Rights (OPEN O20/14776480).

Algorithmic processes run faster and are more ubiquitous than human processing capabilities. If privacy were a right to be guaranteed to users of digital technologies, we need to understand its specific scope, motivation and eligible trade-offs. If privacy were a value with which we need to align those algorithmic processes, we need to specify it mathematically to the level that we can construct an algorithm that detects whether privacy is violated. Our intended contribution is to use logic specifications both as a goal but also as a method to clarify the distinction between different concepts of privacy. Our aim in this project is to make privacy specifications accessible for algorithmic analysis. Only with the precision of logic specification we can compare two privacy conceptualisations and know whether they refer to the same or different ideas.

In this paper, we have aimed for formalization using a multi-modal logic in which we can accommodate the basic principles of *some* of the different approaches, definitions of the right to privacy and then reason with them. Since we are interested in the different deontic consequences of each approach, for a logical-legal theoretical background, we use the theory of normative positions. Our aim is to show the variety with formal conceptual analysis, we do not provide meta-logical results. First we look into some privacy definitions, then shortly into the theory of normative positions (readers being familiar with the latter can skip that section). After those, first we introduce the language and then we provide formal representations of the different rights included or implied by the privacy approaches.

2 Approaches to and Definitions of Privacy

In this section we outline briefly the relation between the idea of privacy and how it is reflected different "computational" domains where privacy is discussed. We then discuss some definitions of privacy that have been influential in the past in law and social science and which we choose to focus on in our specification efforts later in this paper. As it will be obvious from the definitions we consider, we limit ourselves to privacy that a person can enjoy with respect to information about one self. In addition to informational privacy, one can discuss spatial privacy, bodily privacy, privacy of decisions etc.

Privacy is colloquially equated with the concern of how personal data is handled[3]. This is particularly the case in the context of data processing, including collection. Privacy is a long known and studied concept in cybersecurity. The field of *differential privacy* [Dwork, 2008], for example, is concerned with methods for sharing data sets without making individuals identifiable in them. The perception of privacy as concern for how personal data is handled sometimes also "bleeds" into the field of artificial intelligence, where also sometimes is equated to security issues regarding data access [Liu et al., 2021].

It is not particularly clear in the literature what impact AI directly has on privacy if we expand beyond the data security concerns. AI-constructed behavior

[3] for a definition of personal data see [GDPR, 2016]

prediction helps identify patterns in private and, what we can call personal, data [Slavkovik et al., 2021]. [Ackerman et al., 2001] emphasize that "privacy is maintained by allowing the user to disseminate only the necessary data, which cannot be used to identify the user". It can be argued, however, that it is the data collection itself and the use of the analysis done by AI that directly contribute towards reducing the users' rights to control which information is available to whom and whose scrutiny is allowed. What machine learning does is find patterns in data. Data patterns allow us to infer information that is not explicitly available, and which might be information that someone is unwilling to share about themselves. AI, as it is today, does not erode privacy as part of its operation. It is *how AI is used* rather than *what it does*—that is the issue at hand.

Given that we are interested in the normative space of actions that privacy implies, we investigate what the *right to privacy* means. Definitions of what elements this right contains and what duties it entails vary and have developed over the years in different contexts [Matzner and Ochs, 2019]. Privacy can be seen as the right to be left alone, or to be exempt from unwanted scrutiny [Rössler, 2005]—or, as often referred to in US case law, freedom from unwarranted publicity[4]—or, for instance, to be exempt from social interaction [Schwartz, 1968]. It is widely recognized that privacy is both beneficial for personal and social development [Margulis, 2003, Rössler, 2005] and affected by the ease of information creation and processing enabled by digitization [Schwartz, 2004]. There are efforts to help the denizens of the digital world to understand the implications of their own activities on their own privacy, however without a consensus on what information is relevant and how it should be communicated [Barth et al., 2022].

Different authors have argued different privacy perspectives over time, and it can be argued that the right to privacy has evolved as we have evolved as a society. It is not our intention in this section to provide a systematic review of all the privacy definitions. We can and do only focus on few privacy definitions.

We start with [Warren and Brandeis, 1890] that provide one of the first and very influential legal definitions of privacy as the "right to be let alone". The definition of [Warren and Brandeis, 1890] comes in the time when photography and printed press begin to impose on people's lives [Matzner and Ochs, 2019]. In the context of information, we can interpret it as the right that certain information about an individual is not accessible to anyone in any circumstances (contexts).

In 2023, one can argue, the modern human owner of a smartphone is never alone. We can connect to other people instantaneously via the internet, but when we do that we leave digital traces that reveal very much about us [Stachl et al., 2020]. Not being alone does not directly mean being without privacy. Already in 1968, [Westin, 1967] proposed that privacy can be seen "as the claim of [individuals] to determine for themselves when, how, and to what extent information about them is communicated to others." This relaxes the privacy definition beyond the simple "no access" to the "no access without permission", namely to

[4] For instance, Hogin v. Cottingham, 533 So. 2d 525 (Ala. 1988) citing Norris v. Moskin Stores, Inc., 272 Ala. 174, 132 So. 2d 321 (1961)

the requirement that access of particular information (in all contexts) is in the hands of the person who is the subject of that information.

[Nissenbaum, 2009] argues that this idea of access with control is problematic. Information about one self can be freely permitted under some circumstances but not others. For example, while one can be happy to disclose one's HIV status in a dating app, the same permission cannot be taken to hold outside of that specific context[5], for example to hiring agents and insurance companies even if the app is open to everyone. In addition to access and control, she would also specify context. The privacy requirement becomes access with permission in a particular context for a particular aim.

On a somewhat orthogonal dimension [Rössler, 2005] argues that what is problematic about access and sharing of information is based on what one is allowed to do with that information. Namely, the problem lies in using information about someone to stigmatise and scrutinise that individual who should have had the right to privacy. The motivation from this consideration comes from constraints that prevent someone to be alone in their private activities such as for example a disability or limitation of available resources (space, time, funds etc.). Therefore [Rössler, 2005] argues that privacy can be understood as a 'space' where one can act without unwanted public scrutiny. The purpose of affording this freedom from public scrutiny is to preserve the autonomy and freedom of the individual.

We are working with *private information* that we rather loosely—and only informally—define as information about an individual which that individual is not comfortable with being collected, processed, shared, known, used, accessed etc. by others. Private information overlaps, but may not necessarily subsume personal data as defined in the [GDPR, 2016]. Private information in this sense is not only subjective but also contextual: the same information can be private in one context (for some people) but not in another. Within this work we abstract from the context details for the purpose of building up to capturing differences between the different strengths of epistemic privacy requirements.

Lastly we should clarify that influential taxonomies of privacy do exist, although somewhat dated, such as the one of [Solove, 2006]. Solove bases his taxonomy on activities that invade privacy. Activities that invade privacy are arguably easier to discern by a human judge that wants to determine if privacy is violated. However we are concerned with privacy erosion that occurs because of the contemporary capabilities of data science and AI. To this end we focus on the epistemic aspects of privacy and we take an epistemic approach to our specifications in logic.

The *first step* on the road leading to enabling specification is presented in this paper where we use a multi-modal logic for the *formal conceptual characterization* of possible approaches to what the right to privacy means.

[5] The dating app Grinder was fined 6.2 million euros in 2021 for such a violation https://www.forbrukerradet.no/side/grindr-hit-with-e-6-2-million-fine-in-response-to-complaint-from-the-norwegian-consumer-council/.

3 Rights in the Theory of Normative Positions

As a background framework, we use the theory which aims at formally differentiating in the different types of rights: the theory of normative positions [Sergot, 2013]. Its origin is the paper of W.N. Hohfeld who found that lawyers overuse the word 'right' meaning different concepts without even reflecting on it [Hohfeld, 1923]. To resolve this terminological and thus conceptual confusion, Hohfeld proposed to differentiate the following four types of rights and their correlative duties (for details see [Markovich, 2020]):

A claim-right of an agent concerns the counter-party's actions. The counter-

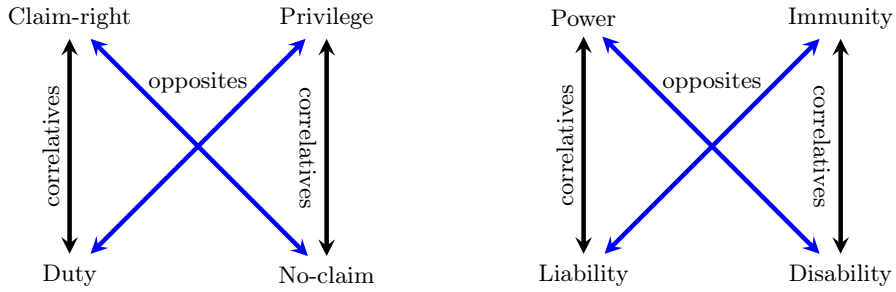


Fig. 1. The Hohfeldian atomic types of rights, and their correlatives

party has an obligation to do the certain thing, and this obligation is directed to the right-holder. Hohfeld calls this a duty, in the narrow sense. The freedom or privilege[6] to do something, on the other hand, is understood as not being the subject of a claim-right coming from the counter-party. Privilege can thus be seen as a directed version of the standard (weak) permission in deontic logic.

The normative positions in the right square capture the agent's ability to change an (other) agent's normative positions. For that reason, they have been called higher-order or *capacitative* [Fitch, 1967]. They thus capture the norm-changing potential—or in case of disability, the lack thereof—of an agent [Dong and Roy, 2021, Markovich, 2020]. The counterparty's exposedness to this change is called liability in this theory, while immunity is the type of right when there is no such exposedness since the other agent has no power.

The theory of normative positions covers the tradition of aiming at formalizing these positions established by the work of Kanger and Kanger (e.g.[Kanger and Kanger, 1966, Kanger, 1972]) and Lindahl (e.g.[Lindahl, 1977]), and later joined by many (e.g. [Makinson, 1986, Sergot, 2013, Sartor, 2005, Markovich, 2020, Dong and Roy, 2021])

[6] 'Freedom' is an often used alternative for 'privilege' in the literature dealing with Hohfeld.

3.1 Rights as Absolute Positions

One of the main characteristics of the theory of Hohfeld is that it considers agents pairwise, as [Makinson, 1986] put it: it is inherently relational. This aspect is often source of criticisms allegedly lead a narrow scope: considering agents pairwise is a good tool to describe contractual situations, but inadequate for the so-called absolute positions (like, for instance, the ownership). This critique is ill-founded. Hohfeld's famous essay on the fundamental legal conceptions had a second part in which he differentiates between the *paucital* and *multital* rights in the case each right-type. A paucital right-relation refers to situations in which we indeed have one-one agent on each side of the relation, like in the case of contracts. Multital rights are, however, series of such relations where one agent takes the right-holder positions and everyone else is a duty-bearer regarding her rights. As the example of [Simmonds, 2001] shows: "I am the owner of Blackacre. I have a claim-right that you should not enter the land without my consent. I have the identical claim-right against your mother, my employer, the Bishop of Ely, and anyone else that you care to mention. Each of these claim-rights is a consequence of my ownership of Blackacre. These are 'multital rights'." The same is true for the owner's freedom to walk on his own land: it means that no one has a claim against him to refrain from walking through it, which is a multital freedom; the owner has a multital power to sell this land: everyone is exposed to the change this sale bring in their normative positions; and the owner has a multital immunity too as everyone else is disable to sell his land changing his normative positions about it.

This addition has great relevance in using this theory for modeling privacy rights: many of the rights privacy implies seems to refer to a unique position that we have against everyone else. We will represent it as a conjunction of relations between one agent and each of the others on the (given) set of agents.

3.2 Rights as Molecular Normative Positions

Hohfeld's initiative didn't succeed in the sense that people—lawyers too—still use the word 'right' without special reflection to what exactly they mean. From the analysis of the attached regulation one can mostly figure out which normative position is covered by the expression. For instance, as pointed out in [Markovich and Roy, 2021b], in Hungary the citizens have a right to know the declaration of assets both in the case of MPs and the local representatives. But these two rights are different: the actual regulation orders the declarations of the MPs to be submitted and made public, while in the case of the local representatives, the law says that in case of a citizen's inquiry, the representative is obliged to submit her declaration to be made public. That is, the first right is a claim-right, the second is a power. However, rights are often refer to not even one atomic position, that is, one of the above mentioned four right positions, but a molecular one. The broader the context, the more probable case is that the right we talk about is a complex one. In case of the human rights, establishing some fundamental interests and their protection, this is rather probable. See, for

instance, the logical analysis of freedom of thought [Markovich and Roy, 2021d]. We believe that the right to privacy is often understood as a combination of different atomic rights positions.

3.3 Epistemic Rights

The notion of epistemic rights as a robust category and investigation of them within the Hohfeldian theory was established in epistemology by [Watson, 2021]. Watson condsiders epistemic rights as those protecting and governing the distribution and accessibility of epistemic goods. Developing (and extending existing) logics for formalizing epistemic rights within the theory of normative positions was initiated by [Markovich and Roy, 2021d] and [Markovich and Roy, 2021c]. In these papers we find a differentiation between epistemic rights in the narrow and the broad sense referring to the theory of normative positions. According to this, epistemic rights in the narrow sense are those that concern the right-holder's epistemic state, like the right to know or the freedom of belief. In the broad sense though, those rights are also epistemic rights that concern the duty-bearer's epistemic state, like the right to be forgotten and the different rights to privacy. Hence, in this paper, we investigate the formalization of these latter as epistemic rights within the theory of normative positions.

4 Language and Semantics

For the formal characterization of these epistemic rights, we use a combination of standard deontic logic augmented with directed operators [Markovich, 2020], action, epistemic, and alethic logic. We are going to work with the following multi-modal language:

Definition 1. *Let A be a finite set of agents and Φ a set of propositional letters. The language $\mathcal{L}$ is defined as follows:*

$$p \in \Phi \mid \phi \wedge \psi \mid \neg\phi \mid \Box\phi \mid \{E_a\phi \mid O_{a\to b}\phi \mid K_a\phi\}_{a,b\in A}$$

$\mathcal{L}$ thus extends the propositional logic with four modalities. E_a is the agency modality and should be read as "agent a sees to it that...". $O_{a\to b}$ is a directed obligation modality, and should be read as "agent a has a duty towards b that...". K_a, on the other hand, is an epistemic modality, to be read as "agent a knows that...". The $\Box$ is the alethic modality "it is necessary that". All these modalities have duals: the weak permissions operator, i.e. $P_{a\to b}$..., which stands for $\neg O_{a\to b}\neg$...; $\langle K_a\rangle$... which stands for $\neg K_a\neg$...; and $\Diamond$..., which stands for $\neg\Box\neg$....

We make the following assumptions regarding the logical behavior of these modalities. We take the deontic modalities $O_{a\to b}$ to be normal modalities validating the D axiom, i.e. $O_{a\to b}\phi \to P_{a\to b}\phi$. So the deontic fragment of our language is standard deontic logic. In this paper we work with a very weak action logic, so we take the agentive modalities E_a to be non-normal, validating only substitution under logical equivalence and the T axiom ($E_a\phi \to \phi$). The

epistemic modality K_a is assumed to be normal modality validating the T,4,5 (and thus the B) axioms, that is, we choose K_a to be the standard knowledge modality. The $\Box$ operator refers to a universal modality satisfying also K,T,4, and 5. Given these assumptions, the language $\mathcal{L}$ is interpreted over frames containing a neighborhood function for each E_a, a deontic ideality relation for each $O_{a\to b}$, an epistemic accessibility relation for each K_a and an alethic accessibility relation.

Definition 2. *Let A be a finite set of agents. A frame $\mathcal{F}$ is a tuple of the following form:*

$$\mathcal{F} = \langle W, R^{\Box}, \{f_a, R^K_a, R^O_{a,b}\}_{a,b\in A}\rangle$$

Here W is set of possible worlds. The function $f_a : W \to \wp\wp(W)$ is a neighborhood function such that, for all $w \in W$ and $X \in f_a(w)$, we have $w \in X$. $R^O_{a,b} \subseteq W^2$ is a serial binary relation. $R^K_a \subseteq W^2$ and $R^{\Box} \subseteq W^2$ are Euclidean relations. A model $\mathcal{M}$ is a frame $\mathcal{F}$ together with a valuation function $V : \Phi \to \wp(W)$.

With this in hand the truth conditions of formula of our language is defined in the standard way. We have only defined explicitly the case for the modalities.

Definition 3. *Let $||\phi|| = \{w : \mathcal{M}, w \models \phi\}$. Then:*

- $\mathcal{M}, w \models \Box\varphi \Leftrightarrow \forall w'(wR^{\Box}w' \Rightarrow \mathcal{M}, w' \models \varphi)$
- $\mathcal{M}, w \models E_a\varphi \Leftrightarrow ||\phi|| \in f_a(w)$
- $\mathcal{M}, w \models \mathbf{O}_{a\to b}\varphi \Leftrightarrow \forall w'(wR^O_{a,b}w' \Rightarrow \mathcal{M}, w' \models \varphi)$
- $\mathcal{M}, w \models K_a\varphi \Leftrightarrow \forall w'(wR^K_a w' \Rightarrow \mathcal{M}, v \models \varphi)$

These truth conditions are standard for the normal modalities K_a, $O_{a\to b}$, and $\Box$. The agency operator E_a is given the so-called *exact* neighborhood semantics [Pacuit, 2017]. Validity in models, frames, and classes thereof, are defined as usual. Since we do not make any specific assumptions regarding the interaction between these modalities, the set of validities over our intended class of frames is completely axiomatized by all propositional tautologies, the logic ET for the agentive modality E_a, KD for $O_{a\to b}$, and S5 both for K_a and $\Box$.

4.1 Motivation of the Language

We chose to use this language to be able to express different variants of what the right to privacy might mean. The directed obligation refers to the Hohfeldian duty emphasizing the relationality, which will always have an E operator in its scope (however, for a staring point, we show below formulas with an undirected obligation too). The very weak action logic enables to talk about "actions" in a very broad sense and even iterate the operator (which would not be so easy with a usual S4 or S5 STIT logics) without engaging with the substantial questions of what actually actions are. We chose S5 though to 'knowledge'. We are aware that the adequate choice of axioms for properly characterizing knowledge is extensively discussed, and we do not intend to take position with our choice. At

this phase of the current research project we put the emphasis of the finding the formulas expressing the variant of the rights related to privacy[7] Using the different combinations we intend to express some basic components of (privacy-related) actions and positions, such as $\Diamond E_a\phi$ as an ability to make it the case that ϕ, $\Diamond E_a O_{b\to a} E_b\phi$ as having the power to put a duty on b to make it the case that ϕ. The formula $\Diamond K_a\phi$ is intended as a has access to ϕ, $E_b\Diamond K_a\phi$ as b making ϕ accessible for a as opposed to $E_b K_a\phi$ as b making a know ϕ. The modularity of the combinations enables us to express seemingly only slightly different concepts which however might have very different consequences. To have a simple language and since we always operate with a finite set of agents, we choose to stay in propositional modal logic.

5 Formalization of Some Right to Privacy Definitions

5.1 Right to be left alone: the right to control who has access

To say that agent a has to right to make it the case that others (b such that $b \neq a$) do not know some information (ϕ)—as in it should be the case—seems to be a legit starting point:

$$O\Diamond E_a\neg K_b\phi \tag{1}$$

It is somewhat different to say is that agent a has to right to make it the case that others cannot know (do not have access to) some information:

$$O\Diamond E_a\neg\Diamond K_b\phi \tag{2}$$

The two formulas above are 'ought-to-be' formulas, they do not express rights directly. In order to fit the theory of normative positions the agents of the normative relations have to be specified. Formally this can be done with the obligation operator being indexed with a pair of agents as introduced in [Markovich, 2020]. In order to have 'ought-to-do' formulas, an action operator have to be in the argument of the obligation operator indexed with the obligation's first indexed agent. The obvious candidate for creating such a situation is the state (legislator). It seems to be plausible to say for some specified set of formulas, it is the state's duty to make it the case that an agent can decide about the publicity of ϕ. Actually, if we accept that it is a state duty, then it is regarding each of its citizen:

$$\bigwedge_{a,b\in A} O_{s\to a} E_s\Diamond E_a\neg\Diamond K_b\phi \tag{3}$$

Actually, this requirement might be too strong toward the state. The legislator's tool is rulemaking, not implementing technical constraint (not to mention metaphysical ones). Thus it seems to be more appropriate to say the following:

$$\bigwedge_{a,b\in A} O_{s\to a} E_s O\Diamond E_a\neg\Diamond K_b\phi \tag{4}$$

[7] In a later phase of this research, we will modify the logic according the findings, such as counterintuitive consequences in a given context, using different modalities as the epistemic notions involved in the discussion about privacy might greatly vary.

Such a legislation does not solve the problem yet as it does not identify the agent which has to make it the case. We need to point out a duty-bearer:

$$\bigwedge_{a,b\in A} O_{s\to a}E_sO_{c\to a}E_c\Diamond E_a\neg\Diamond K_b\phi \tag{5}$$

The agent $c \in A$ can be a company, or any other agent that the state can impose such a duty on, where we also allow for $c = b$ (but we require $a \neq b$ and $a \neq c$). In the Hohfeldian terms, the formula above is a *claim-right* of (every) a against to state to establish a clam-right against the relevant company making it possible that a can decide who knows ϕ. According to the interpretation of [Westin, 1967] privacy is attained by a person when that person can control who can share and use their information thus including another related claim-right of a realized by a duty of everyone else to refrain from making a unable to let others know:

$$\bigwedge_{b,c\in A} O_{c\to a}\neg E_c\neg\Diamond\neg E_aK_b\phi \tag{6}$$

The right to privacy definitely includes a's *multital freedom* as well: that she does not have an obligation letting others know about ϕ (or that she even makes ϕ accessible to others):

$$\bigwedge_{b\in A} \neg O_{a\to b}E_aK_b\phi \tag{7}$$

$$\bigwedge_{b\in A} \neg O_{a\to b}E_a\Diamond K_b\phi \tag{8}$$

However, a freedom this way, in itself, is just a weak permission. It has to come with some protection to relaize what we usually mean by what a freedom is. The classical protection is what the Hohfeldian *immunity* covers: the disability of other to change this freedom, which looks like the following in our formalization:

$$\bigwedge_{b\in A} \neg\Diamond E_bO_{a\to b}E_a\Diamond K_b\phi \tag{9}$$

In the above cases we use the tacit assumption that it is possible that someone can be left alone in the metaphorical sense of having total control on the access to ϕ. However, this is not always the case.

As [Rössler, 2005] argued, being alone or access to control to private information may be unattainable in some circumstances or for some people. In such a case, we have to calculate with agents who do have access to ϕ, and the rights to privacy are realized in some control in the normative space of these agents regarding ϕ-related actions. So in cases where it is inevitable that b has access to ϕ, one obvious candidate for a's privacy rights is that a can prohibit (or permit) making ϕ :

$$\bigwedge_{b,c\in A} \Box((\Box\Diamond E_bK_b\phi)\to(\Diamond E_aO_{b\to a}\neg E_b\Diamond K_c\phi)) \tag{10}$$

It can be questioned whether in these situations it is indeed b who sees to it that he knows ϕ, thus the formula below might be found more accurate:

$$\bigwedge_{b,c \in A} \Box((\Box\Diamond K_b\phi) \to (\Diamond E_a O_{b\to a} \neg E_b \Diamond K_c \phi)) \tag{11}$$

It is an interesting interpretation of privacy as space without uninvited scrutiny to say that agent a might not only want that further agents have no access to ϕ but maybe she has to be able to control whether she has to face uninvited opinions of those who necessarily have access to ϕ. That is, even if b as a helper or servant is necessarily witnesses ϕ, a's right to privacy covers that she prohibits that b lets her know about this:

$$\Box((\Box\Diamond K_b\phi) \to (\Diamond E_a O_{b\to a} \neg E_b K_a \phi)) \tag{12}$$

Note that this formula differs from the one above only in missing a $\Diamond$ and talking about b letting know a and not a third party.

5.2 Right to transparency

In some systems, the question is not really the total exclusion of any type of access (as it might not be feasible under all circumstances), but rather the right to transparency: agent a should know about whether anyone has access to ϕ. To express such a claim-right, that is, the directed obligation of the agent controlling the system, we apply the solution of [Hulstijn, 2008] of 'knowing whether' which avoids the infamous Åqvist's paradox [Åqvist, 1967] making it possible at the same time that we do not rely on conditionals:

$$\bigwedge_{b \in A} O_{c\to a}(E_c K_a \Diamond K_b \phi \vee E_c K_a \neg \Diamond K_b \phi) \tag{13}$$

5.3 Protection: possibility of enforcement

An important aspect of our claim-rights, that is, the duties of others regarding our privacy is that once they are violated, we (on the metaphysical level) have a new claim-right against the judiciary to enforce or rights (or compensation for the violation) as described in detail and formalized in [Markovich, 2020]. For instance, in case of the company's duty to enable the user to make ϕ inaccessible:

$$\Box((\neg E_c \Diamond E_a \neg \Diamond K_b \phi) \to O_{j\to a} E_j E_c \Diamond E_a \neg \Diamond K_b \phi) \tag{14}$$

This (on the practical level) means that we have a *power* to initiate a legal action putting a duty on the judiciary to decide whether indeed that was the case what we state. This instrumental aspect is discussed in detail and formalized in [Markovich and Roy, 2021a], we do not go into details here.

6 Discussion, Related and Future Work

We have introduced a multi-modal logic to formalize some approaches of what the right to privacy means pointing out several different normative positions. This work brings together two aspects that have been present in computer science. On the one hand, the need for expressing privacy-related concepts have been addressed in the literature using logic for (legal) knowledge representation. [Aucher et al., 2011] and [Aucher et al., 2010], in order to deal with privacy policies, investigated both the obligation and the permission to know, differentiating between obligatory and permitted knowledge and obligatory and permitted messages. [Li et al., 2022] use dynamic logic to describe permitted announcements. On the other hand, within the context of multi-agent systems, privacy is studied from several aspects such as: artificial agents assisting people in maintaining their privacy [Kökciyan and Yolum, 2022], identifying "leaks" of particular information [Dennis et al., 2016], negotiation to resolve privacy conflicts among people [Such and Rovatsos, 2016, Kekulluoglu et al., 2018], preservation of privacy during learning [Nagar et al., 2021], to name a few. In these approaches, privacy is seen as different property of states and/or actions, but not as an epistemic right. Both in logic and MAS, the works by different authors build on different understandings of privacy—our work aims exactly at making them comparable. Furthermore, since these considerations of privacy in logic and MAS are typically not grounded in the law and social sciences literature, it is also difficult to ground the them into the state-of-the-art outside of computer science.

Our work aims at providing foundations for a research going for implementable specifications of privacy-related rights. This paper provides and initial formal conceptual analysis contributing to legal knowledge representation, and to set a basis in which to ground privacy work AI, MAS, including policy modeling, policy-as-code and law-as-code paradigms and initiatives. Among our next steps there are addressing the defeasibility of these rights, trying other formalisms, and using the LogiKEy framework for the design and engineering of ethical reasoners, normative theories and deontic logics put forth by [Benzmüller et al., 2020] to see which works best.

References

Ackerman et al., 2001. Ackerman, M., Darrell, T., and Weitzner, D. J. (2001). Privacy in context. *Human - Computer Interaction*, 16(2-4):167–176.

Aucher et al., 2010. Aucher, G., Boella, G., and Torre, L. V. D. (2010). Privacy policies with modal logic: the dynamic turn. In *DEON 2010*, pages 196–213.

Aucher et al., 2011. Aucher, G., Boella, G., and van der Torre, L. (2011). A dynamic logic for privacy compliance. *Artificial Intelligence & Law*, 19(2/3):187 – 231.

Barth et al., 2022. Barth, S., Ionita, D., and Hartel, P. (2022). Understanding online privacy? A systematic review of privacy visualizations and privacy by design guidelines. *ACM Comput. Surv.*, 55(3).

Benzmüller et al., 2020. Benzmüller, C., Parent, X., and van der Torre, L. (2020). Designing normative theories for ethical and legal reasoning: Logikey framework, methodology, and tool support. *Artificial Intelligence*, 287:103348.

Dennis et al., 2016. Dennis, L. A., Slavkovik, M., and Fisher, M. (2016). "How did they know?" Model-checking for analysis of information leakage in social networks. In Cranefield, S., Mahmoud, S., Padget, J. A., and Rocha, A. P., editors, *XII - COIN 2016*, volume 10315 of *LNCS*, pages 42–59. Springer.

Dong and Roy, 2021. Dong, H. and Roy, O. (2021). Dynamic logic of legal competences. *Journal of Logic, Language and Information*, pages 1–24.

Dwork, 2008. Dwork, C. (2008). Differential privacy: A survey of results. In Agrawal, M., Du, D., Duan, Z., and Li, A., editors, *Theory and Applications of Models of Computation*, pages 1–19, Berlin, Heidelberg. Springer Berlin Heidelberg.

Elvy, 2017. Elvy, S.-A. (2017). Paying for privacy and the personal data economy. *Columbia Law Review*, 117(6):1369–1459.

Fitch, 1967. Fitch, F. B. (1967). A Revision of Hohfeld's Theory of Legal Concepts. *Logique et Analyse*, 10(39/40):269–276.

Floridi and Cowls, 2019. Floridi, L. and Cowls, J. (2019). A unified framework of five principles for ai in society. *Harvard Data Science Review*, 1(1).

Forbrukerrådet, 2018. Forbrukerrådet, N. (2018). Artificial intelligence and privacy.

GDPR, 2016. GDPR (2016). General data protection regulation 2016/679 of the European Parliament and of the Council.

Hohfeld, 1923. Hohfeld, W. (1923). Fundamental legal conceptions applied in judicial reasoning. In *Fundamental Legal Conceptions Applied in Judicial Reasoning and Other Legal Essays*, pages 23–64. Yale.

Hulstijn, 2008. Hulstijn, J. (2008). Need to know: Questions and the paradox of epistemic obligation. In van der Meyden, R. and van der Torre, L., editors, *DEON 2008*, volume 5076 of *LNCS*, pages 125–139. Springer.

Kanger, 1972. Kanger, S. (1972). Law and logic. *Theoria*, 38:105–132.

Kanger and Kanger, 1966. Kanger, S. and Kanger, H. (1966). Rights and parlamentalism. *Theoria*, 32:85–115.

Kekulluoglu et al., 2018. Kekulluoglu, D., Kökciyan, N., and Yolum, P. (2018). Preserving privacy as social responsibility in online social networks. *ACM Transactions on Internet Technology (TOIT)*, 18(4):42:1–42:22.

Kinnier et al., 2000. Kinnier, R. T., Kernes, J. L., and Dautheribes, T. M. (2000). A short list of universal moral values. *Counseling and Values*, 45(1):4–16.

Kökciyan and Yolum, 2022. Kökciyan, N. and Yolum, P. (2022). Taking situation-based privacy decisions: Privacy assistants working with humans. In Raedt, L. D., editor, *Proceedings of IJCAI-22*, pages 703–709.

Li et al., 2022. Li, X., Gabbay, D., and Markovich, R. (2022). Dynamic Deontic Logic for Permitted Announcements. In *Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning*, pages 226–235.

Lindahl, 1977. Lindahl, L. (1977). *Position and Change—A Study in Law and Logic*. Synthese Library. D. Reidel, Dordrecht.

Liu et al., 2021. Liu, B., Ding, M., Shaham, S., Rahayu, W., Farokhi, F., and Lin, Z. (2021). When machine learning meets privacy: A survey and outlook. *ACM Comput. Surv.*, 54(2).

Makinson, 1986. Makinson, D. (1986). On the formal representation of rights relations: Remarks on the work of Stig Kanger and Lars Lindahl. *Journal of Philosophical Logic*, 15(4):403–425.

Margulis, 2003. Margulis, S. T. (2003). Privacy as a social issue and behavioral concept. *Journal of Social Issues*, 59(2):243–261.

Markovich, 2020. Markovich, R. (2020). Understanding Hohfeld and Formalizing Legal Rights: the Hohfeldian Conceptions and Their Conditional Consequences. *Studia Logica*, 108.

Markovich and Roy, 2021a. Markovich, R. and Roy, O. (2021a). Cause of action and the right to know. In *Legal Knowledge and Information Systems - JURIX 2021 Frontiers in Artificial Intelligence and Applications 346*, pages 217–224. IOS Press.
Markovich and Roy, 2021b. Markovich, R. and Roy, O. (2021b). Formalizing the right to know: Epistemic rights as normative positions. In *Logics for New-Generation AI*, page 154.
Markovich and Roy, 2021c. Markovich, R. and Roy, O. (2021c). Formalizing the right to know: Epistemic rights as normative positions. In *LNGAI 2021*, pages 154–159.
Markovich and Roy, 2021d. Markovich, R. and Roy, O. (2021d). A logical analysis of freedom of thought. In *DEON 2021*.
Matzner and Ochs, 2019. Matzner, T. and Ochs, C. (2019). Privacy. *Internet Policy Review*, 8(4).
Nagar et al., 2021. Nagar, A., Tran, C., and Fioretto, F. (2021). Privacy-preserving and accountable multi-agent learning. In *Proc. of the 20th AAMAS*, page 1605–1606.
Nissenbaum, 2009. Nissenbaum, H. (2009). *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. Stanford University Press, Stanford.
Pacuit, 2017. Pacuit, E. (2017). *Neighborhood semantics for modal logic*. Springer.
Quine, 1978. Quine, W. V. (1978). *On the Nature of Moral Values*, pages 37–45. Springer Netherlands, Dordrecht.
Åqvist, 1967. Åqvist, L. (1967). Good samaritans, contrary-to-duty imperatives, and epistemic obligations. *Nous*, 1(4):361–379.
Rössler, 2005. Rössler, B. (2005). *The Value of Privacy*. Polity, Cambridge.
Sartor, 2005. Sartor, G. (2005). *Legal Reasoning. A Treatise of Legal Philosophy and General Jurisprudence*. Springer.
Schwartz, 1968. Schwartz, B. (1968). The social psychology of privacy. *American Journal of Sociology*, 73(6):741–752.
Schwartz, 2004. Schwartz, P. M. (2004). Property, privacy, and personal data. *Harvard Law Review*, 117(7):2056–2128.
Sergot, 2013. Sergot, M. (2013). Normative Positions. In Gabbay, D., Horty, J., Parent, X., van der Meyden, R., and van der Torre, L., editors, *Handbook of Deontic Logic and Normative Systems*, pages 353–406. College Publications.
Simmonds, 2001. Simmonds, N. (2001). Introduction. In *Hohfeld: Fundamental legal conceptions as applied in judicial reasoning*, Classical Jurisprudence series. Ashgate, Aldershot, new ed. / edited by David Campbell and Philip Thomas. edition.
Slavkovik et al., 2021. Slavkovik, M., Stachl, C., Pitman, C., and Askonas, J. (2021). Digital voodoo dolls. In Fourcade, M., Kuipers, B., Lazar, S., and Mulligan, D. K., editors, *AIES '21: Conference on AI, Ethics, and Society*, pages 967–977. ACM.
Solove, 2006. Solove, D. J. (2006). A taxonomy of privacy. *University of Pennsylvania Law Review*, 154(3):477–564.
Stachl et al., 2020. Stachl, C., Au, Q., Schoedel, R., Gosling, S. D., Harari, G. M., Buschek, D., Völkel, S. T., Schuwerk, T., Oldemeier, M., Ullmann, T., Hussmann, H., Bischl, B., and Bühner, M. (2020). Predicting personality from patterns of behavior collected with smartphones. *Proceedings of the National Academy of Sciences*, 117:17680–17687.
Such and Rovatsos, 2016. Such, J. M. and Rovatsos, M. (2016). Privacy policy negotiation in social media. *ACM Trans. Auton. Adapt. Syst.*, 11(1).
Warren and Brandeis, 1890. Warren, S. D. and Brandeis, L. D. (1890). The right to privacy. *Harvard Law Review*, 4(5):193–220.
Watson, 2021. Watson, L. (2021). *The Right to Know. Epistemic Rights and Why We Need Them*. Routledge.
Westin, 1967. Westin, A. F. (1967). *Privacy and Freedom*. Atheneum, New York.

Privacy issues on applications of AI

Vilmos Gábor Rádi

Károli Gáspár University of the Reformed Church, 1042 Budapest, Viola street 2-4., Hungary
dr.radi.vilmos.gabor@gmail.com

Abstract. Artificial intelligence is a term often bandied about in the media, and for the average person it is still a science fiction concept that comes to mind in movies. The reality is that artificial intelligence (and the machine learning that underpins it) is already part of our everyday lives; we all encounter it several times a day, unconsciously, when using a smartphone or social media, when shopping online, or even without any visible signs of it, such as in the case of a secret facial recognition software.
The legal area of data protection is closely linked to the evolution of technology, with new innovative technologies, in particular so-called 'disruptive' technologies, raising new data protection issues and risks. Artificial Intelligence sets up challenges for legal thinking, and there are many scientists who have elaborated on possible rules about the feasibility of AI's legal personality; issues of the legal responsibility, or IP rights concerning machine learning. However, the field of privacy is especially concerned, because the lifeblood of machine learning is the same thing what privacy tries to protect: your personal data. In my study I am making an attempt to discover some risks and possible mitigations.

Keywords: Artificial Intelligence, Privacy, Data Protection.

1 Introduction

Machine learning has been around for circa a half century now; however, it became a cultural phenomenon counting on the attention of the wider public only in the last 10-**15 years**.. However, machine learning has a different nature than many other formally regulated products or services. The specific nature of machine learning and artificial intelligence requires a unique approach from the regulator - or from a possible supervisory authority as well.

Data protection has been present for some time now as a minor area of law; but it **became relevant only with the advancement of automated electronic registers**. As the computing capacity increased, more and more data – including personal data – could be collected, stored, and used more effectively than before. It quickly became obvious **that these datasets can be used to create larger databases, combine formerly stand-alone information to meaningful context**, which can be used for various purposes by the owner of the database. This reality has boosted the need for an effective regulation which protects peoples' rights and dignity. In recent years, data protection / privacy

regulations have been issued in many countries worldwide. The legal regulations usually have similar approach about what could be expected from the owner / user of the database.

Since machine learning and artificial intelligence has a unique operational method, the **data protection issues these softwares are raising are of an unusual and different nature,** which implies that a novel approach is needed to address them. In order to take a closer look at the problem, I will attempt to discover the specific data protection issues raised by the technology, and then address the specific issues raised by its application in specific sub-areas. I expect that the realities of machine learning and **AI will stretch and test the traditional conceptual framework of data protection;** compliance with data protection requirements will be challenging for the actors involved;

As machine learning is used more and more in the applications people use every day, the **need for legally regulation, or at least setting up ethical frameworks** or preparing guidelines by relevant authorities became an ever more pressing matter

but I **envisage that compliance with data protection requirements can be achieved in the use of AI, based on the already known principles and methods in legal regulations, combined with novel solutions**.

For the sake of easy and global understating, **I will use the most well-known data protection framework, the General Data Protection Regulation (hereinafter: GDPR)** [1], in my study.

I am of the firm belief that regulation is swiftly needed, and I shall attempt to not only showcase why AI poses a danger to personal data, but how I, as a senior privacy expert would go about solving the issues it presents.

2 Roles of the parties in case machine learning processes of personal data

If we are talking about any kind of legal relationship, the first thing to do is examine who are the relevant parties involved in the process. According to the logic of the GDPR, the role of the **data controller** can essentially be understood as **deciding** on the collection of personal data, the scope of the data collected, the goals it will be used for, and the way it will be used.

If several organisations decide together on the use of the data, we speak of **joint controllers**,

and if several organisations cooperate in a process and use personal data, **but decide separately on their use,** we speak of parallel, **independent controllers**.

It is also important to mention the role of the **data processor, who carries out specific processing operations** on the basis of instructions from the controller. Finally, remember the concept of the recipient: the controller/processor transmits personal data to the recipient.

How do the roles evolve if we think of the **developer** of the AI / in the case of a **robot**, its **manufacturer** /---------- the **distributor** of the AI or robot / ----------the **organisation** using or operating the AI?

The developer of the AI is not necessarily the same as the distributor, nor even the market actor that produced the robot.

It is possible that both the robot manufacturer and the company that developed the AI - which is different from the robot manufacturer – will use the data collected during the machine learning process, and the operation of the software; it is possible that they decide jointly on the scope of the data (in which case they are joint controllers) or unilaterally on the basis of the instructions of the robot manufacturer (in which case the AI developer becomes the data processor).

It is also possible that they decide separately on the scope of data to be used, e.g. the robot manufacturer specifies that it needs 47 types of data, but the AI developer stipulates in their contract that it can use these data for its own purposes, OR independently of the contract, **for further development; or it may collect 18 additional data in addition to the 47 dat**a; in this case the two actors are also parallel, independent data controllers.

If we look at the data management from the perspective of **the company using the AI or robot, this company may be the data controller, the robot manufacturer the data processor and the AI developer the sub-processor**; but if the robot manufacturer or the AI developer also uses other data for their own purposes, then again, they are also parallel, independent data controllers for these data.

The world of automated vehicles shows a potentially interesting scenario. For the sake of deeper understanding, here I will provide a little technical distinction between solutions used for self-driving: the **Singular Self-Driving Vehicle (SSDV) method versus the Linked Self-Driving Vehicle (LSDV)** solution. In the SSDV method, all the units involved in the traffic use only the **information detected by their own sensors and take action based on that information**. In the LSDV solution, the transport units do not only use the data they collect, but also receive information from a **large number of sensors in the environment and incorporate environmental information as a basis for automated decisions**. In this case, it is more rational to have a central system that controls individual vehicles in relation to each other and to the vehicles that are in traffic but not connected. [2] So, who can be the data controller in transport automation? In addition to the considerations already mentioned above, in the case of LSDV solutions, it is conceivable that a single central AI could manage the traffic, but not on behalf of the **company operating the fleet**, but maintained by, for example, a **municipality or the state itself**. In this case, the municipality/state becomes the data controller as it determines the purposes.

We still have a very important actor: the data subject. Who is this person? Generally speaking, the data subject is the natural person, whose data will be collected and used during the machine learning process, and later on, during the operational phase of the software / robot, when it uses information from its environment for it's working. But who is this person exactly? If we know the context of privacy, it is essential that we should know who can be affected by the collection of data. Well, sometimes it can be obvious but sometimes can be surprising and not foreseeable. For example, if we are talking about a chatbot, the data subject will be the user **who interacts with the chatbot**. Let's see another example, if we are talking about artificial intelligence used for **marketing purposes, everyone who is in that database**, will be a data subject – just think about **Facebook**, which also uses AI, eventually all Facebook users qualify as data subjects. Similarly, every person who uses a **search engine**, became data subject, since even the search parameters are important information to the software. If we look in an even larger scale, in case of an artificial intelligence working for governmental bodies, the software possibly will have access to **national registers**, so every citizen of that country will be data subject, indifferently whether they want it or not. But let's see some example in a minor scale: automatically analyised footages from **body cameras**. In this case, the wielder of the camera is obviously a data subject, however, everyone else coming into front of this person will be recorded and eventually became affected with this data processing; possibly even without knowing about being recorded.

Let's visit again our complex example, the case **of automated vehicles.** The driver inside the car will be a data subject even if the vehicle is not self-driving; the speed, usage of the brakes, **general driving style** is often monitored and collected in modern cars. The more advanced the car, the more information it collects; for example, it can detect the **tiredness or sleepiness of the driver**. However, the driver is not the only person affected. If other persons in the **passenger** compartment are monitored in any way and can be identified (e.g. by a passenger compartment camera), they are also concerned. The data subject category is really opened up by **on-board cameras** and **cameras mounted on the outside of the vehicle**, as they can record practically anyone on the street, so the number of potential data subjects is practically infinite. Again these persons potentially do not know they images will be processed. Let us not forget the **drivers of other vehicles, wh**ose data will also be processed in case of a **Linked Self-Driving Vehicle system**. Finally, although automation is not necessary, it is important to remember that if the data from a phone call passes through the vehicle's systems and is stored or transmitted in any way, the other party to the call will also be affected.

We can see that even defining the role of actors under GDPR can lead to complicated scenarios. However, the role of each party can be deduced, if we are taking into consideration all of the affected persons.

3 Data protection principles: purpose limitation, data minimization and accuracy

The principles of purpose limitation, data minimisation and accuracy referred to in the GDPR arise in relation to the quantity and quality of data input into machine learning software. In the context of AI, one would think that the more data developers put into the software, the more efficient and "smarter" the Artificial Intelligence will be and the more accurate the conclusions it will draw.

However, practice has shown that the opposite is true, and that AI **fed by large amounts of low-quality data can often come to the wrong conclusions**, and can also lead to **discriminatory AI**. An example of this **is Tay,** the AI software developed by Microsoft, which has been given its own account on Twitter in order to learn by communicating with users. However, the amount of information it received from Twitter users, without any prior filtering, very quickly led to Tay becoming a prejudiced, racist program, and also calling for sex acts with profanity. As a result, Twitter decided to suspend the artificial user just sixteen hours after Tay's account was activated. [3]

While this case is an interesting example, the **White Paper of the European Commission on Artificial Intelligence** states that "**human decision-making is not free from error and bias**. However, the same bias present in **artificial intelligence can have a much greater impact**, adversely affecting and differentiating many people without the social control mechanisms that guide human behaviour." [4]

In addition to the striking example above, there are countless cases that demonstrate that "AI developers also face the problem of exactly what data the machine should learn from, and how to obtain appropriately cleaned and structured data that can be used for learning. The performance of any learning algorithm or AI built on top of it can only be as good as the quality of the data used to teach it (...) legal problems (e.g. discriminatory decision making) arise if the data tables used for machine learning are of poor quality or poorly assembled (...) machine learning is only as good as the data used." [5]

When it comes to the data used for machine learning, it is more important to collect pre-selected, high-quality data than quantity. **Finding good quality data also intersects with the principle of *accuracy*.**

It is important to specify the application domain of the AI before collecting the data and **to collect only relevant data (*purpose limitation principle*).**

The principle of ***data minimisation*** **also applies here**: it is important to determine how much data is needed for effective machine learning, and to first enter a limited amount of test data and then check whether the software is efficient enough to draw accurate conclusions. Depending on the result of the check, a decision should be taken whether or not to enter additional data; entering unnecessary data would violate the principle of data minimisation, somewhat on a par with storing data only for later use, and building a data inventory just for its own sake. [6] Eventually, as we see, these 3 basic principles of data protection can be complied with during the operation of machine learning softwares.

De-anonymization serious risk!!

4 The transparency principle

Applying the principle of transparency to AI is a challenge, as the results produced by AI are **not always clearly explainable or predictable, such as the black box phenomenon** described above. This creates a challenge in terms of how to create AI that works transparently as required by the GDPR, and how to comply with the obligation to provide information on the logic used and the information that can be understood about it, in the case of information and access requirements as referred to in Articles 13-14-15 of the GDPR.

The networked nature of the AI, the **number of layers** and the **complexity** of the relationships between them make it difficult to present in a way that is understandable to the average person. As the GDPR requires substantive information on the logic used, it is not enough to simply state that the decision is based on AI, as even the data subject **will not be able to understand** the basis on which the decision is made, and the use of **complex descriptions and jargon** also makes it difficult to provide transparent and understandable information.
Furthermore, a detailed disclosure of the algorithm is not necessary, as this would also **affect the intellectual property rights** of the AI developers.
It may also be a challenge if the controller is not fully aware of the logic used.

A solution to this dilemma could be **not to share explicitly the details** of the algorithm with the data subject, but to provide information on **how a change in the input data will result in a change in the 'output'.**

Information should then be provided on the **significance of the processing and the likely consequences for the data subject**.

This is particularly important where the processing has a legal or similarly significant impact, e.g. the

- processing of credit applications,
- -the determination of the "waiting list position" of a patient on a health waiting list,
- the impact of software used in the justice system, etc.

A good solution could be to **publish a comparative application, a test system**, where data subjects can experiment with test data to see what the results are for them if they enter other data into the software.

Another type of transparency problem if the **data subjects not realizing at all if they are dealing with an artificial intelligence**. For example, in case of chatbots, it is often not clear to users that they are not dealing with a human being (especially when

human names are given to chatbots to improve the user experience). Transparency requires that this is made clear and that the organization which put the chatbot in its webpage should inform the users about the processing of data by the chatbot.

A particular danger is that some chatbots programmed to be attackers can be embedded in instant messaging applications (e.g. Viber, WhatsApp, messenger, etc.) where they can contact the unsuspecting user and **pose** as a traditional chatbot, e.g. a pizza ordering chatbot**. This phenomenon is called smishing.** If the user reacts and gives out information about themselves, they may become a victim of data theft similar to traditional phishing [7]

A similar issue can arise with **mass surveillance** using **CCTV cameras combined with face recognition softwares**.

People walking across the streets

- may not even realize the well-placed CCTV cameras;
- and the problem will get worse if they not aware about the existence of the face recognition technology behind the system, which is operated by an artificial intelligence.

While the usage of face recognition technologies in itself is differently evaluated in different countries, we can say at least, if the operator of the surveillance system wants to keep a high standard of privacy, it should

a) first, notify the people about the CCTV system, and

b) second, provide detailed information about the face recognition technology used with the surveillance system. This information sheet should be easily accessed, and should also provide contact details to the operator, so the affected persons can exercise their rights.

In the context of the transparency requirement, **the Cambridge Analytica** case is worth mentioning. This case is important not because of the black box effect, but it shows the significant influence of machine learning softwares, affecting even democratic structures. In the early 2010s, Aleksandr Kogan, a researcher at Cambridge University, **developed an application** for Facebook called "**This is Your Digital Life"** (TIYDL for short), which created a **psychological profile** of its users. The app is made under Facebook Platform Policy rules. The app also sought **consent from each user concerned.** It later emerged that the **app did not only access the data of the data subjects using it but also their friends.** The data compiled on the user and their friends psychological profile of the user and his/her friends included their political orientation, what content or actors they follow on Facebook, what their attitude to religion and where they rank on the so-called **OCEAN scale which is an acronym for the English names of five attributes (Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism).** [8]

Kogan had transferred the data to third parties, including Cambridge Analytica and Eunoia Technologies. In 2015, Facebook noticed this and removed the app from the site, and asked Kogan and those companies for written confirmation that they had

destroyed the data that had been unlawfully processed. In March 2018, former employee Christopher Wylie claimed that Cambridge Analytica had not destroyed the data, but had actively targeted certain impressionable voter groups with political ads during the 2016 US presidential election campaign, using the profiles previously created in the data to successfully influence those voters in so-called "swing" constituencies.

5 Rights of the affected persons

The privacy regulations usually devote a separate section to the rights of data subjects, because without effective exercise of privacy rights, the whole data protection system does not worth anything.

One of the most fundamental rights of data subjects is the **right to information**, which is reflected on the AI's owner's / operator's side (which we can also call data controller according to the GDPR, as we described above) as an obligation to provide information.

First of all, the **transparency issue** come into play, but **let's suppose we solved this problem** with the remediation technique described in the previous section. Even after this, we can face other difficulties.

Second problem: **HOW to provide in a timely manner?**

Let's return to the **automated vehicles**. The data controller, who is responsible for the artificial intelligence working in the self-driving car or other moving object, is responsible to inform everyone affected by the data processing. The first step of this data processing is creating video footages by the **small cameras placed around the exterior of the car. We** have to take into account the other **road users, including drivers of other cars, pedestrians, people just sitting on bench at the side of the road,** etc.

It is practically **impossible to provide immediate information** that a vehicle passing by is recording; nor is it realistic that the QR code on the side of the car could be read by someone while driving (in fact, it would be dangerous). This leaves the publication of a privacy notice as the one suitable, classic solution.

Also one of the most fundamental rights is the **right to erasure**. This may confront with the **most essential need of the artificial intelligence: the need for data**. If we delete the information already implanted into the software, during its learning phase, can we delete it without harming the capability of the software to recognize patterns, or suggest decisions?

The problem can be even more complex: if we look at the fact that **some of the data u**sed in the context of traffic automation can be **interpreted independently** (e.g. technical status data)

but **other data can only be interpreted in combination (e.g. real-time location data in relation to other road users, camera images, data detected by sensors,** etc.). In the latter case, the request for erasure of data may also affect other natural persons, so it is by no means certain that **this request can be granted in all cases, and a case-by-case assessment is necessary**.

For example, re**ctification or deletion of data concerning other persons could involve the loss of potential evidence in a possible lawsuit.**

It is also worth mentioning that, for example, standards or even legislation may **require full data retrieval at a later stage, for technical reliability.**

The situation is similar where one wishes to exercise **the right of access** - in fact, the right of subsequent communication - or the right of rectification in respect of data **which also concern other persons** as described above, such as other transport operators, in whose case it is questionable how much data can be disclosed about them.

The right to **data portability:** according to the GDPR, the data subject shall have the right to **receive the personal data concerning him or her, which** he or she has provided to a controller, in a structured, commonly used and machine-readable format and have the right to transmit those data to another controller without hindrance from the controller to which the personal data have been provided. If we want to comply with a data subject request regarding data portability,

we **will face the following problem:** we may provide the information of the specific situation (for example a credit request), on which the artificial intelligence based the specific decision, but in itself, **without the underlying algorithm, the dataset probably have no additional value for the person**.

If another artificial intelligence will get the same dataset, that AI probably uses a different decision making method, and will reach a different conclusion based on the same dataset. A possible solution for this issue could be, if the data controller operating the AI provides not just the dataset in a machine-readable format, but also the very basic description of the logic behind the decision, and probably a test system, as described in Section 4.

If we examine the automated vehicles once again, in the case of **the LSDV system,** if the whole traffic is controlled by a **single central AI, there is presumably no other similar data controller than the one who controls the system, so that the exercise of the data portability right becomes pointless;** perhaps if we think of a lawsuit by a natural person in connection with a traffic accident, the release of data might be relevant there.

6 Human intervention

Pursuant to Article 22 (1) of the GDPR, it is also necessary to a**llow the data subject to opt out of decisions based solely on automated processing**, including profiling, which would have **legal effects concerning him or her or similarly significantly affect** him or her, and to request human intervention pursuant to Article 22(3). Human intervention is relevant if it could lead to a different result from the decision taken by the algorithm. In order to do so, the person acting must examine the data used in the process relevant to the outcome of the algorithm and must independently carry out a balancing exercise on the basis of which he or she must formulate his or her own decision. **Human intervention can also help to detect and filter out discriminatory automatic decision-making.**
The White Paper of the European Commission also identifies as one of the risks that AI decisions are **sometimes difficult to challenge effectively** - hence the case for maintaining the possibility of human intervention.

Take a look again at automated vehicles. A request by the data subject not to be covered by automated data processing would in this case **effectively involve the disabling of some - or even all - of the services of the vehicle**. It is also worth bearing in mind that some processes of the vehicle may be interlinked, so **if the data subject requests the cessation of certain automated functions, this may also entail the cessation of other automated vehicle functions.**

The dilemma is **even more acute in the case of an LSDV system** where **the whole traffic is controlled by a central AI**; in this case, the request of the data subject implies a complete exit from the system.

Without knowing the future LSDV systems that will be implemented, **this may imply, for example, a change of insurance or individual legal consequences for operators who join or leave the LSDV system**, as may be foreseen in future transport legislation.

The request for human intervention by the controller is difficult to interpret in the case of transport automation; perhaps this could include a **passenger/driver in a vehicle requesting a real person to take control of the vehicle by remote control**. This would raise further questions, such as whether this would be considered sufficiently safe.

It is also worth mentioning that the WP29 Working Party of national supervisory authorities (the predecessor of the European Data Protection Board before the GDPR), WP249, Opinion No 2/2017 on data management at the workplace, document 5.7. The Recommendation states that "the legitimate interest of the company in monitoring its drivers does not, however, take precedence over the rights of those drivers to the protection of their personal data." [9]

7 Conclusion

With regards to my hypothesis, **specific privacy issues** related to AI have been presented, the **most serious of which is the black box phenomenon**, which makes it very difficult to meet the transparency criterion.
But another a problem is that **some non-personal data may become personal data through the use of algorithms (because the algorithm is already able to associate these data with a natural person).**

Furthermore, the **identification of data controllers and processors** is also problematic in some places; the reasonable expectations of data subjects often do not cover the practical use of AI, i.e. they are not aware of the purposes for which their data are used or even of the processing itself;

and AI allows or facilitates **the monitoring and even profiling of large numbers of people**.

Further problems may arise in some areas of AI application, e.g. **enforcing the right to erasure may be difficult** in chatbots or traffic automation; information may be difficult to implement in self-driving cars, as it is not practical to put information or even QR codes on the cars;
and **some technologies may be explicitly intrusive, such as face recognition technologies, o**r even ore in case of facial emotion recognition technologies, especially for employees who may not have the possibility to refuse to wear these devices.

It can be seen that AI is pushing the boundaries of privacy and challenging actors. However, a **consistent but novel use of privacy principles can address these challenges, for example, by describing the proposals in the case of AI explainability** (not explicitly sharing details of the algorithm with the data subject but providing information on how a change in the data input will result in a change in the 'output'; and explaining the relevance of the method and the expected consequences for the data subject. The author of this study considers the hypothesis to be well-founded, since, on the basis of a review of the subject, he believes that the application of the principles of **privacy-by-design and privacy-by-default, the carrying out of impact assessments, th**e focal points described below can provide a reassuring solution to the problems that arise.

Focal points for the lawful use of AI:

(a) the existence of an **adequate, legitimate and acceptable legal basis** for the processing,
(b) the application of the principle of **data minimisation** and the inclusion of only the necessary data in the software,

(c) **ensuring transparency**; showing how different data can produce different results in the software, rather than complex technical specifications; even providing a test system may be the best way to achieve this,

d) the **possibility of human intervention** should be ensured in all cases, with a particular focus on whether human judgement can produce a different result from that of the AI. [10]

These are the issues that need our immediate attention. In order to try to solve the data protection problems arising out of AI's rapid spreading and lack of legislation, my humble opinion is that these focal points should be kept at the forefront of legislators. The coming of the EU's draft legislation is not enough to stop data breaches worldwide. Rather, there should be a **dialogue worldwide between states and private companies**, platforms, so that AI, which is here to change our world forever, is properly handled. I advise **using the method of human rights dialogue in order to ensure that the difference between how various countries implement data protection regulations**, as this will undoubtedly be the next biggest change in human history.

References

1. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance) https://eur-lex.europa.eu/eli/reg/2016/679/oj, last accessed 2023/04/17.
2. Udvary, S.: Technológia és jog kölcsönhatása a közlekedés megújulásában (The interaction of technology and law in the renewal of transport), Conference presentation. Győr, Széchenyi István University, Deák Ferenc Faculty of Law and Political Science, (2017)
3. Schwartz, O.: In 2016, Microsoft's Racist Chatbot Revealed the Dangers of Online Conversation, IEEE Spectrum, https://spectrum.ieee.org/in-2016-microsofts-racist-chatbot-revealed-the-dangers-of-online-conversation, last accessed 2023/04/17.
4. White Paper on Artificial Intelligence: a European approach to excellence and trust, https://ec.europa.eu/info/sites/default/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf, last accessed 2023/04/17.
5. Karácsony G.: Okoseszközök – Okos jog? (Smart gadgets - smart law?) Regulatory issues of artificial intelligence, Dialóg Campus, (2020)
6. Eszteri, D.: Hogyan tanítsuk jogszerűen a mesterséges intelligenciánkat? (How to train legally our artificial intelligence?) Magyar Jog, 66 (12). pp. 669-682. ISSN 0025-0147 (2019)
7. Hasal, M.J., Nowaková J., Ahmed K., Hussam, S., Václav, A., Ogiela, S.: Chatbots: Security, privacy, data protection, and social aspects https://onlinelibrary.wiley.com/doi/full/10.1002/cpe.6426, last accessed 2023/04/17.
8. Ackerman, C.E.: MA., Big Five Personality Traits: The OCEAN Model Explained, https://positivepsychology.com/big-five-personality-theory/, last accessed 2023/04/17.
9. 29 Article Working Party - Opinion 2/2017 on data processing at work - wp249, https://ec.europa.eu/newsroom/article29/items/610169/en, last accessed 2023/04/17.
10. Eszteri, D.: Hogyan tanítsuk jogszerűen a mesterséges intelligenciánkat? (How to train legally our artificial intelligence?) Magyar Jog, 66 (12). pp. 669-682. ISSN 0025-0147 (2019)

Author Index