Arguments, Rules and Cases in Law: Resources for aligning learning and reasoning in structured domains

Cor STEGING [a], Silja RENOOIJ [b], Bart VERHEIJ [a] and Trevor BENCH-CAPON [c]
[a] *Bernoulli Institute of Mathematics, Computer Science and Artificial Intelligence, University of Groningen*
[b] *Department of Information and Computing Sciences, Utrecht University*
[c] *Department of Computer Science, University of Liverpool*

Abstract. This paper provides a formal description of two legal domains. In addition, we describe the generation of various artificial datasets from these domains and explain the use of these datasets in previous experiments aligning learning and reasoning. These resources are made available for the further investigation of connections between arguments, cases and rules. The datasets are publicly available at `https://github.com/CorSteging/LegalResources`.

1. Introduction and context

The resources described here are made available for the further investigation of connections between arguments, cases and rules. This topic is gaining increased relevance in artificial intelligence in general, in particular by the widespread acceptance that explainable approaches to machine learning are needed, with argumentation-based machine learning as a relevant new angle of research [3,28]. In particular, the learning of knowledge used for reasoning in structured domains remains a relevant topic of research. Here we focus on two example domains in the field of law.

Argumentation in AI & Law is typically based on two kinds of sources: legislation and precedents. Concretely, legislation often provides the grounding for the arguments used in rule-based reasoning, and precedents for those used in case-based reasoning. However, in actual legal reasoning, many hybrid combinations appear (cf. also research on the comparison of various legal systems [10,11]), which has inspired AI & Law research on hybrid rule-based/case-based approaches already for a long time (e.g., [16]). The datasets and domains that we describe can be used to investigate a variety of approaches including neural networks [4,18], rule mining [8,9,23] and inductive logic programming [13], or other relevant techniques.

2. Resource description

We artificially generate datasets based on two legal domains: a fictional Welfare benefit domain and the tort law domain. Both domains are defined by clear knowledge struc-

Table 1. An overview of the two domains. Listed are the number and type of features, the number and type of conditions to be learned, whether or not all cases are covered by the datasets (complete) and whether the domain is fictional or real.

	Features		Conditions			
Domain	**no.**	**type**	**no.**	**type**	**Complete**	**Fictional**
Welfare benefit	64	Boolean & numerical	6	independent	No	Yes
Tort law	10	Boolean	5	dependent	Yes	No

tures. These datasets contain instances with various features and an output label that is determined by these features as defined by the knowledge structure of the domain. All instances in the datasets are therefore valid in the sense that their output labels follow from evaluating the knowledge structure that defines the domain. An overview of the two domains alongside their key characteristics can be found in Table 1, which are explained in the upcoming sections.

2.1. Welfare benefit domain

The Welfare benefit domain was first introduced by Bench-Capon in an experiment investigating whether neural networks can learn rules from data [4]. It has later been used in several applications (see Section 3) including argument based machine learning [13] and argumentation dialogue based on association rules [23]. It is a fictional legal domain that concerns the eligibility of a person for a welfare benefit to cover the expenses for visiting their spouse in the hospital. It is defined by six conditions:

1. The person should be of pensionable age (60 for a woman, 65 for a man);
2. The person should have paid contributions in four out of the last five relevant contribution years;
3. The person should be a spouse of the patient;
4. The person should not be absent from the UK,
5. The person should have capital resources not amounting to more than £3,000;
6. If the relative is an in-patient the hospital should be within a certain distance: if an out-patient, beyond that distance.

These are meant to represent a variety of types of conditions found in such benefits. They were also expected to present a range of challenges to the neural networks in [4]. In order of expected difficulty the conditions can be seen as examples of the following functions:

- A positive Boolean function (3);
- A negative Boolean function (4);
- A numeric threshold function (5);
- A symmetric Boolean function, where a certain number of variables need to be true, in no particular order (2);
- A numeric threshold function with the threshold dependant on another feature (1);
- A numeric XOR function with the polarity dependant on another feature (6).

These conditions can be formalised as follows:

$$Eligible(x) \iff C_1(x) \wedge C_2(x) \wedge C_3(x) \wedge C_4(x) \wedge C_5(x) \wedge C_6(x)$$

Table 2. Features in the Welfare benefit domain.

Feature	Values
Age	0 – 100 (all integers)
Gender	male or female
$Con_1, \ldots, Con_5$	true or false
Spouse	true or false
Absent	true or false
Resources	0 – 10,000 (all integers)
Type (Patient type)	in or out
Distance (to the hospital)	0 – 100 (all integers)

$$C_1(x) \Longleftrightarrow (Gender(x) = female \wedge Age(x) \geq 60) \vee (Gender(x) = male \wedge Age(x) \geq 65)$$
$$C_2(x) \Longleftrightarrow \|Con_1(x), Con_2(x), Con_3(x), Con_4(x), Con_5(x)\| \geq 4$$
$$C_3(x) \Longleftrightarrow Spouse(x)$$
$$C_4(x) \Longleftrightarrow \neg Absent(x)$$
$$C_5(x) \Longleftrightarrow \neg Resources(x) \geq 3000$$
$$C_6(x) \Longleftrightarrow (Type(x) = in \wedge Distance(x) < 50) \vee (Type(x) = out \wedge Distance(x) \geq 50)$$

Using these conditions, we can generate artificial datasets. The six independent conditions for eligibility are defined in terms of 12 variables, which are the features of the generated datasets. These features and their possible values are shown in Table 2. Note that one can easily change the the upper and lower bounds of the integer values of the features in our source code. In [4], a further fifty two irrelevant noise features were added to discover whether the neural net could identify the twelve relevant features and thus sort the wheat from the chaff.

To generate a Welfare benefit dataset using our code, one must specify three function parameters: the number of instances, the number of noise features and the label distribution. By default, the number of noise features is set at 52, just as in [4], yielding a total of 64 features plus an eligibility label for each instance. These noise features have integer values ranging from 0 to 100, unrelated to eligibility. By default, exactly half of the instances in these datasets are eligible, creating a balanced label distribution, as is common practice in machine learning experiments.

For the eligible instances, feature values are generated (randomly where possible) such that they satisfy the conditions $C_1 - C_6$. For each condition, $\frac{1}{6}$th of all of the ineligible instances are designed to fail on that specific condition; where possible the values of the features involved are generated randomly such that the condition fails. While this might not necessarily lead to a realistic distribution, it does provide a uniform distribution of the conditions. Each condition is therefore responsible for the ineligibility of $\frac{1}{6}$th of the ineligible instances. All remaining features in these instances are generated randomly across their full range of values (see Table 2); as a result, it is possible for ineligible instances to fail on multiple conditions, and some conditions will fail more often than others. Alternatively, using an additional function parameter, it is possible to generate datasets where ineligible instances fail on only a single condition. Changing the datasets so that they fail on a single feature has been shown to improve the behavior of machine learning models in rationale evaluation tasks [4,18,19]. More details regarding the effects of this variation can be found in the original publications.

2.2. Tort law domain

Our second domain concerns a fragment of the real life legal domain of Dutch tort law. Articles 6:162 and 6:163 of the Dutch civil code describe when an action is wrongful and resulting damages must be repaired [20]:

> **Art. 6:162 BW.** 1. A person who commits an unlawful act toward another which can be imputed to him, must repair the damage which the other person suffers as a consequence thereof. 2. Except where there is a ground of justification, the following acts are deemed to be unlawful: the violation of a right, an act or omission violating a statutory duty or a rule of unwritten law pertaining to proper social conduct. 3. An unlawful act can be imputed to its author if it results from his fault or from a cause for which he is answerable according to law or common opinion.
>
> **Art. 6:163 BW.** There is no obligation to repair damage when the violated norm does not have as its purpose the protection from damage such as that suffered by the victim.

The arguments and attacks regarding this 'duty to repair' (*dut*) is visualized Figure 1 [20] and can be further formalised as follows:

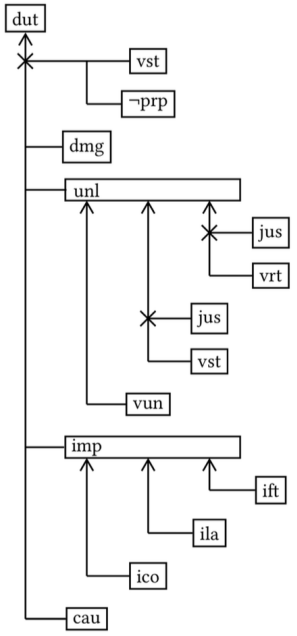
$$
\begin{aligned}
&dut(x) \iff c_1(x) \wedge c_2(x) \wedge c_3(x) \wedge c_4(x) \wedge c_5(x) \\
&c_1(x) \iff cau(x) \\
&c_2(x) \iff ico(x) \vee ila(x) \vee ift(x) && \textit{(unl)} \\
&c_3(x) \iff vun(x) \vee (vst(x) \wedge \neg jus(x)) \vee (vrt(x) \wedge \neg jus(x)) && \textit{(imp)} \\
&c_4(x) \iff dmg(x) \\
&c_5(x) \iff \neg(vst(x) \wedge \neg prp(x))
\end{aligned}
$$

Here the elementary propositions are provided alongside an argumentative model of the law in Figure 1 [20], and conditions c_2 and c_3 capture the legal notions of unlawfulness (*unl*) and imputability (*imp*), respectively.

The Dutch tort law domain is captured in 5 conditions for duty to repair (*dut*), based upon 10 Boolean features. Each condition is a disjunction of one or more features, possibly with exceptions. The feature capturing a violation of a statutory duty (*vst*) is present in both condition c_3 and c_5, rendering these dependent. Note that the abstract notions of unlawfulness (*unl*) and imputability (*imp*) are not features but conditions.

The tort law domain with its 10 Boolean features captures $2^{10} = 1024$ possible unique cases that can be generated from the argumentation structure of the tort law domain in Figure 1. Each case has a corresponding outcome for *dut*, indicating whether or not there is a duty to repair someone's damages.

To generate a tort law dataset using our code, one must specify two function parameters: the number of instances and the label distribution. By default, datasets of the tort law domain are generated such that *dut* is true in exactly half of the instances. They are generated by sampling uniformly from all unique cases, such that each possible case is represented equally within the given label distribution. Note that the Tort law domain only contains 1024 unique cases and therefore datasets with more than 1024 instances are guaranteed to contain duplicates.



(A) Arguments and their attacks in the domain of Dutch tort law.

dut	There is a duty to repair someone's damages.
dmg	Someone has suffered damages by someone else's act.
unl	The act was unlawful.
imp	The act can be imputed to the person that committed the act.
cau	The act caused the suffered damages.
vrt	The act is a violation of someone's right.
vst	The act is a violation of a statutory duty.
vun	The act is a violation of unwritten law against proper social conduct.
jus	There exist grounds of justification.
ift	The act is imputable to someone because of the person's fault.
ila	The act is imputable to someone because of law.
ico	The act is imputable to someone because of common opinion.
prp	The violated statutory duty does not have the purpose to prevent the damages.

(B) Elementary propositions in the domain of Dutch tort law.

Figure 1. Arguments and attacks (A) and their elementary propositions (B) in Dutch tort law [20].

3. Resource use

The datasets described above were used to introduce a method for evaluating and potentially improving the decision-making of machine learning models [18] and to illustrate the utility of the method in a set of experiments [17]. Machine learning models were trained on the datasets and tasked with predicting eligibility in the Welfare benefit domain and a duty to repair damages in the Tort law domain. These trained models were then investigated to examine whether their decision-making matched the knowledge structures that defined the domain. The datasets were also used in follow-up experiments wherein the method for rationale evaluation was compared to explainable AI techniques [19]. These experiments used additional datasets for which further details can be found in the original publications.

3.1. Previous use of the Welfare benefit domain

The Welfare benefit domain was introduced in [4] to investigate neural networks in problems of open texture. The aims were to discover whether a neural net could accurately predict the outcome of legal cases represented as feature vectors without any guidance from domain knowledge, and more importantly to see whether it would apply the correct rationale in predicting these outcomes. This meant using a dataset where the rationale was known, and so a dataset was generated from a set of rules. The question was then whether the neural net would correctly discover these rules. The results showed that

while performance was good, neither the pensionable age nor the distance conditions were satisfactorily recognised. The relevant features were mostly identified, although two of the irrelevant features were accorded more significance than sex, distance and patient status, reflecting the inability to discover the last two conditions. Following its creation for [4], the dataset was made available to and reused in several subsequent projects.

An experiment to determine whether association rules [1] could be mined from a set of legal cases represented as feature benefits was described in [5]. This exercise used the Welfare benefit dataset from [4]. The algorithm used to mine the rules [8] worked only on Boolean data. Hence the data was pre-processed to assign the numeric features to two or more bins. Where available, the ranges for these bins was determined using domain knowledge, thus age was either less than 60, 60-64, or greater than 64. If the knowledge is not available, a number of arbitrary ranges could be used as in [26]. The pre-processing also stripped out the irrelevant features. Even with the pre-processing, success was only partial. In particular the condition relating to distance and patient status presented difficulties.

[9] describes HeRo, a greedy, best-first, branch-and-bound algorithm designed to induce defeasible logic theories from large datasets, similar to Inductive Logic Programming [15] algorithms designed to produce standard Horn clause theories. The paper included comparisons with other approaches and previous work, including [4], which used the original dataset. The resulting theory was said to achieve a high degree of accuracy, but did not contain any reference to the condition regarding the paid contributions. An interesting feature was that it gave award of benefit as a default, with sufficient conditions for non award, rather than six necessary conditions for award of benefit, as originally stated in [4]. This approach of looking for a reason to withhold benefit rather than determining that all the required conditions are satisfied may well be a better approach.

In Argument Based Machine Learning (ABML) [12,13], a standard rule induction algorithm (CN2 [7]) is augmented with arguments from an expert to explain why misclassified cases fail. In this experiment. like [9], the most problematic condition was the contributions condition. However, after six misclassified cases had been explained, a set of rules giving a very high accuracy was achieved, the only blemish on the rules being a slightly inaccurate threshold for the distance condition. A feature of this work was that it also investigated the effect of some items in the dataset being incorrect—an ever present possibility in the Welfare benefit domain where there is often a high error rate in the actual decisions [23]. The experiments, which modified the dataset by changing the classifiction in a set proportion of cases, showed that ABML is in fact highly robust in the face of incorrect data.

The idea behind Arguing from Experience [23,25] is to mine arguments for and against a classification from a data set, and then to deploy these arguments in a dialogue to refine them and then determine which classification should win. The arguments were based on association rules, and the moves in the dialogue on argument moves in case based reasoning: cite, distinguish and counter example (e.g., [2]). This approach was applied to a variety of classification problems, including the Welfare benefit dataset. It operated both on a single dataset (PADUA [23]) and multiple datasets, to represent discussion between people with different sets of examples (PISA [24]). Strategies for deploying the moves were also proposed and evaluated. The project reported high accuracy and, like [13], high tolerance to a proportion of incorrect information.

[14] addressed the problem of finding explanations for a collection of cases where an explanation is a labelled argumentation graph consistent with the cases, and a case is represented as a statement labelling. The Welfare benefit dataset was used in two experiments to evaluate the approach.

3.2. Previous uses of the Tort law domain

In [22], Dutch tort law was used as a case study of the modeling of argumentation in a realistic setting. The study focused on analyzing aspects of informal legal arguments and showing their connections to logical tools.

Dutch tort law was also used as a case study to show the formal connections between arguments, rules, and cases in [20]. The rules of the Dutch tort law domain play the role of knowledge in knowledge-based AI, and cases that of examples in data-driven AI. A case model was developed based on the rule-based arguments and attacks in Dutch tort law, illustrating how statutory, rule-based law can be formalized in terms of cases. The formalization that we use in creating our tort law datasets is based on the arguments and defeating circumstances described in [20].

In [21], the claim is made that we need to study AI as law in order to achieve trustworthy, social, responsible, humane, and ethical AI. It is argued that the solutions proposed in the field of AI & Law, such as argumentation, schemes and norms, rules and cases, have the potential to support the development of good AI in other applications as well. The Dutch tort law domain was used to illustrate the connection between knowledge-based AI and data-driven AI.

Rule-based, case-based and argument-based reasoning are explored in [27]. The relationship between these three major types of modeling legal reasoning are investigated and illustrated using the Dutch tort law domain.

In [6] a dataset was created for US rather than Dutch tort law. Cases from Illinois tort law were translated from natural language into predicate representations with the specific aim to create a domain representation and associated datasets to be used in AI research.

4. Availability

All of the datasets were artificially generated and can be generated again for future research. Jupyter notebooks that illustrate and explain the data generation process, alongside a few example datasets, can be found in a publicly accessible Github repository.[1] Additionally, three example datasets are available as CSV files:

1. `WelfareFailMany2000.csv` contains 2000 cases of Welfare Benefit domain: 1000 eligible cases and 1000 ineligible cases. Ineligible cases fail on at least one, but possibly several, of the conditions.
2. `WelfareFailOne2000.csv` contains 2000 cases of the Welfare Benefit domain: 1000 eligible cases and 1000 ineligible cases. Ineligible cases fail on only one of the conditions.
3. `Tort1024.csv` contains all 1024 unique cases of the Tort Law domain.

[1] `https://github.com/CorSteging/LegalResources`

References

[1] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, pages 207–216, New York, NY, USA, 1993. ACM, New York.

[2] V. Aleven. 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] H. Ayoobi, M. Cao, R. Verbrugge, and B. Verheij. Argumentation-based online incremental learning. *IEEE Transactions on Automation Science and Engineering*, pages 1–15, 2021.

[4] T. Bench-Capon. Neural networks and open texture. In *Proceedings of the 4th International Conference on Artificial Intelligence and Law*, ICAIL '93, pages 292–297, New York, NY, USA, 1993. ACM, New York.

[5] T. Bench-Capon, F. Coenen, and P. Leng. An experiment in discovering association rules in the legal domain. In *Proceedings 11th International Workshop on Database and Expert Systems Applications*, pages 1056–1060. IEEE, 2000.

[6] Joseph Blass and Kenneth Forbus. The Illinois intentional tort qualitative dataset. In *Legal Knowledge and Information Systems - JURIX 2022: The Thirty-Fifth Annual Conference*, volume 362 of *Frontiers in Artificial Intelligence and Applications*, pages 151–157. IOS Press, 2022.

[7] P. Clark and R. Boswell. Rule induction with cn2: Some recent improvements. In *European Working Session on Learning*, pages 151–163. Springer, 1991.

[8] G. Goulbourne, F. Coenen, and P. Leng. Algorithms for computing association rules using a partial-support tree. In *Research and Development in Intelligent Systems XVI*, pages 132–147. Springer, 2000.

[9] B. Johnston and G. Governatori. Induction of defeasible logic theories in the legal domain. In *Proceedings of the 9th international conference on Artificial intelligence and law*, pages 204–213. ACM, New York, 2003.

[10] N. MacCormick and R. S. Summers, editors. *Interpreting Statutes. A Comparative Study*. Dartmouth Publishing, Aldershot, 1991.

[11] N. MacCormick and R. S. Summers, editors. *Interpreting Precedents. A Comparative Study*. Dartmouth Publishing, Aldershot, 1997.

[12] M. Možina, J. Žabkar, T. Bench-Capon, and I. Bratko. Application of argument based machine learning to law. In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 248–249, 2005.

[13] M. Možina, J. Žabkar, T. Bench-Capon, and I. Bratko. Argument based machine learning applied to law. *Artificial Intelligence and Law*, 13(1):53–73, 2005.

[14] R. Riveret. On searching explanatory argumentation graphs. *Journal of Applied Non-Classical Logics*, 30(2):123–192, 2020.

[15] E.Y. Shapiro. The model inference system. In *Proceedings of IJCAI 1981*, page 1064, 1981.

[16] D.B. Skalak and E.L. Rissland. Arguments and cases: An inevitable intertwining. *Artificial Intelligence and Law*, 1(1):3–44, 1992.

[17] C. Steging, S. Renooij, and B. Verheij. Discovering the rationale of decisions: Experiments on aligning learning and reasoning. In *4th EXplainable AI in Law Workshop (XAILA 2021)*, pages 235–239. ACM, 2021.

[18] C. Steging, S. Renooij, and B. Verheij. Discovering the rationale of decisions: towards a method for aligning learning and reasoning. In Juliano Maranhão and Adam Zachary Wyner, editors, *ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021*, pages 235–239. ACM, 2021.

[19] C. Steging, S. Renooij, and B. Verheij. Rationale discovery and explainable AI. In Schweighofer Erich, editor, *Legal Knowledge and Information Systems - JURIX 2021: The Thirty-fourth Annual Conference, Vilnius, Lithuania, 8-10 December 2021*, volume 346 of *Frontiers in Artificial Intelligence and Applications*, pages 225–234. IOS Press, 2021.

[20] B. Verheij. Formalizing arguments, rules and cases. In *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law*, ICAIL '17, pages 199–208. ACM, New York, 2017.

[21] B. Verheij. Artificial intelligence as law. *Artificial Intelligence and Law*, 28(2):181–206, 2020.

[22] B. Verheij, J. Hage, and A.R. Lodder. Logical tools for legal argument: a practical assessment in the domain of tort. In *Proceedings of the 6th international conference on Artificial intelligence and law*, pages 243–249, 1997.

[23] M. Wardeh, T. Bench-Capon, and F. Coenen. Padua: a protocol for argumentation dialogue using association rules. *Artificial Intelligence and Law*, 17(3):183–215, 2009.

[24] M. Wardeh, T. Bench-Capon, and F. Coenen. Pisa—pooling information from several agents: multiplayer argumentation from experience. In *Research and Development in Intelligent Systems XXV*, pages 133–146. Springer, 2009.

[25] M. Wardeh, F. Coenen, and T. Bench-Capon. Multi-agent based classification using argumentation from experience. *Autonomous Agents and Multi-Agent Systems*, 25(3):447–474, 2012.

[26] J. Zeleznikow and A. Stranieri. Knowledge discovery in the split up project. In *Proceedings of the 6th international conference on Artificial intelligence and law*, pages 89–97, 1997.

[27] H. Zheng and B. Verheij. Rules, cases and arguments in artificial intelligence and law. In *Research Handbook on Big Data Law*, pages 374–388. Edward Elgar Publishing, 2021.

[28] K. Čyras, A. Rago, E. Albini, P. Baroni, and F. Toni. Argumentative XAI: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 4392–4399. 2021.