Legal Knowledge and Information Systems
G. Sileno et al. (Eds.)

doi:10.3233/FAIA230945

Improving Rationales with Small, Inconsistent and Incomplete Data

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

Abstract. Data-driven AI systems can make the right decisions for the wrong reasons, which can lead to irresponsible behavior. The rationale of such machine learning models can be evaluated and improved using a previously introduced hybrid method. This method, however, was tested using synthetic data under ideal circumstances, whereas labelled datasets in the legal domain are usually relatively small and often contain missing facts or inconsistencies. In this paper, we therefore investigate rationales under such imperfect conditions. We apply the hybrid method to machine learning models that are trained on court cases, generated from a structured representation of Article 6 of the ECHR, as designed by legal experts. We first evaluate the rationale of our models, and then improve it by creating tailored training datasets. We show that applying the rationale evaluation and improvement method can yield relevant improvements in terms of both performance and soundness of rationale, even under imperfect conditions.

Keywords. Machine Learning, Responsible AI, Explainable AI, Knowledge, Data

1. Introduction

Machine learning systems, such as large language models, are becoming more common due to their increasingly better performance. As a socially disruptive technology, they have the potential to improve many aspects of human life, but can also behave in unexpected, irresponsible ways. These systems can make the wrong decisions, but they can also make the right decisions for the wrong reasons [1]. Ensuring that the decision-making, or rationale, of a system is as desired is a required step towards responsible AI, especially when applied to law.

Many state of the art machine learning models are black boxes that reason without transparency, making it difficult to investigate the rationale of the model. The rising subfield of explainable AI aims to solve this issue by providing explanations for black box models [2]. While explanation techniques can expose the rationale of the AI to a certain extent, they cannot guarantee that the rationale is as desired. Previous work has therefore introduced a hybrid method for evaluating the decision-making of machine learning systems [3], using domain knowledge to test the rationale of such data-driven systems.

Opaque and unsound decision-making is, however, not the only problem with data-driven models in AI & Law [4]. The amount and quality of training data greatly affects

the performance of data-driven AI systems. More, good quality data generally leads to a better performance, but is often difficult to obtain. Due to the changes in the interpretation of the law, inconsistencies can arise. For example, a case that was given a violation verdict twenty years ago, may receive a non-violation if it was tried today, making the older case inconsistent with our current interpretation of the law. Real data is therefore not always consistent, which can be problematic for machine learning models, which are inherently retrospective. Additionally, in many legal cases not all relevant facts are known. Lawyers and judges are able to reason and judge over these cases regardless. Many machine learning systems, however, are not able to deal with missing facts, and require all features to have a value.

In this paper, we investigate how the hybrid rationale evaluation and improvement method performs under such imperfect circumstances. This expands upon earlier work, where the rationale was investigated using artificial data and under perfect conditions [3], i.e. without inconsistencies and missing values, and with sufficient training data (albeit with noise variables that did not contribute to the label). We explore the rationales within the domain of court case predictions. Based on a knowledge structure of Article 6 of the European Court of Human Rights [5], we generate artificial cases on which we train and test our networks. Using the method for rationale evaluation, we can evaluate their rationale and use that evaluation to improve it. In a set of experiments, we investigate how the rationale of the models improves upon varying three different parameters: the training dataset size, the level of inconsistency in the training data, and the amount of missing values in the training data.

2. Domain

Our models are tasked with predicting the outcome of cases from the European Court of Human Rights (ECHR) that deals with cases claiming violations of articles laid out by the European Convention on Human Rights. We focus on Article 6, which is concerned with the right to a fair trial. Cases of the ECHR are readily available online and labelled (violation or non-violation)[1]. Most approaches to court case prediction train a machine learning model on the text of these cases, and task the model with predicting the outcome [6]. Models like BERT or the specialized Legal-BERT variants perform with accuracies of around 80%. Since these large language models are black boxes, it is nearly impossible to extract coherent explanations from them. As a result, the rationale of these type of systems remains largely unexplored.

With the help of legal experts, a knowledge representation of Article 6 was created in the form of an Abstract Dialectical Framework (ADF) [5] using the ANGELIC methodology [7]. This ADF is a hierarchical structure, in which every node has a set of child nodes that determine whether the node should be accepted or rejected. The root node of the ADF for Article 6, represents the *verdict* of a case: violation or non-violation. The children of this verdict node are the *issues* that determine the verdict, as described by legislation. The leaf nodes of this ADF are the *base level factors*, where factors are legally relevant fact patterns. These can be seen as answers to yes-or-no questions that one can have about a case. Intermediary nodes are *abstract factors*, which in turn are

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

defined by either base level factors or other abstract factors. The version of the Article 6 ADF that we use consists of the verdict, 5 issues, 14 abstract factors and 32 base level factors [5]. The verdict is determined by the conjunction of the 5 issues. All five issues I_i therefore need to evaluate to true in order for the verdict to evaluate to 'violation': $Violation(x) \iff I_1(x) \wedge I_2(x) \wedge I_3(x) \wedge I_4(x) \wedge I_5(x)$

Extracting the base level factors from the texts of real-life court cases is non-trivial, and requires legal experts to manually annotate the court cases [8]. Instead, we therefore generate artificial court cases using the Article 6 ADF. Each generated case consists of the 32 base level factors, alongside the verdict of the case as determined by the ADF. The result is a tabular dataset of court cases, which we use to train and test machine learning models.

3. Experimental Set-up

Given the base level factors, the features, our machine learning models will need to predict the verdict, the label. These models are first trained and evaluated under perfect conditions. Then, models are trained and evaluated with limited dataset sizes, inconsistencies and missing values. We evaluate the performance and decision-making of the model using the Tailored Rationale Evaluation and Improvement (TREI) method (from [3]):

1. **Measure the performance** of the trained system using contemporary evaluative measures, and proceed if it is sufficiently high;
2. **Design rationale evaluation test sets** for rationale evaluation, targeting selected rationale elements based on expert knowledge of the domain;
3. **Evaluate the rationale** through the performance of the trained system on these rationale evaluation test sets;
4. **Improve the rationale** if needed, by re-training the system on a tailored training dataset, designed using the results from the rationale evaluation.

While the first step of the method states to only proceed once the performance is sufficiently high, we will proceed to apply the method even if the performance is poor, to investigate whether rationale improvement is possible under imperfect circumstances.

The machine learning models that we use in our experiments are multi-layer perceptrons (MLPs). These relatively simple neural networks are black boxes but sufficient for the tabular, artificial dataset that we use. While other models might perform better, it is not our goal to create the best possible model. Our goal is to investigate the behavior of the neural network under different conditions, and how it can be adjusted using the rationale evaluation and improvement method. For these purposes, the MLPs are sufficient. More specifically, we use a three layered, multi-layer perceptron, implemented using the MLPClassifier of the scikit-learn package [9].

4. Datasets

In this section, we describe the different datasets that we generate and how these are used to train and test our machine learning models. All of the datasets that we create will be made available upon acceptance.

Regular Training Data The regular training dataset contains 5,000 artificial cases, where the base level factors are generated randomly, but such that 50% of the cases in the dataset violate Article 6 and 50% of the cases do not. Since the verdict is defined as a conjunction of all the issues, the cases without violation are generated randomly such that *at least* one issue evaluates to false. There are no further restrictions when generating the dataset. This training dataset is flawless, in the sense that it provides sufficient examples and does not contain any mistakes, inconsistencies or any missing data. In practice, however, training data is often imperfect. We therefore also train on datasets of different sizes, and with inconsistencies and missing values.

Varying Dataset Size To investigate the effects of dataset size on the performance, the rationale and the potential rationale improvement, we generate additional datasets ranging from 10 to 40,000 instances, without any inconsistencies or missing values.

Inconsistencies To evaluate the rationale of the network with inconsistency, we generate special training datasets wherein we alter the cases such that they become inconsistent. There are different ways of generating inconsistency. In this study we choose to flip the values of random features (base level factors) to create inconsistencies. Since all features are Booleans, this means that a 1 (true) will become a 0 (false) or vice versa. The label still remains as it was in the original dataset. Changing the value of a random feature could therefore cause the case to be inconsistent with the ADF, our knowledge representation of the law. For example, a case labelled 'violation' could be made inconsistent by changing the value of factor 'I1F1' from 1 to 0, as Issue 1 is defined by a single factor. Note that by flipping a few features of a case, the case does not necessarily need to become inconsistent. To create inconsistency in our dataset, we flip a percentage of all of the features across all cases. This means that at a 50% inconsistency level, half of the features of all cases are flipped. At a 100% inconsistency level, all features in the data are flipped and and all cases have become inconsistent, effectively generating an inverted version of the ADF. These datasets consist of 50,000 instances without missing values.

Missing Values Incomplete knowledge of all of the factors is common in law. The ADF is able to deal with missing values, as it has a default value when the actual value is unknown. This is not the case for all machine learning models, as most require information about all features. To investigate the networks' rationale under missing values, we generate a training dataset in which we alter cases such that they contain missing values. We select random features and change their values from 0 or 1 to 0.5 to capture the concept of an unknown fact. These cases are still consistent with the ADF since the ADF can deal with missing values. To create datasets with missing values, we select a percentage of all of the features across all cases to contain missing values. This means that at 50%, half of the features of all cases will be missing, and at 100%, all features in the data are missing. These datasets consist of 50,000 instances without inconsistencies.

Training and Testing We train and test neural networks on these different types of training datasets and then evaluate them using a regular test set. This regular test set is generated in the same fashion as the regular training dataset, and also contains 5,000 instances. We first apply the TREI method to a network trained on the regular training dataset, in order to evaluate its rationale and potentially improve it using tailored training datasets based on the evaluation (step 4). We then apply the same TREI method to networks trained on datasets of varying sizes, datasets with inconsistencies or datasets

Table 1. The mean MCC values of the original and tailored networks across 100 runs, trained on 5,000 training instances, on the regular and rationale evaluation test sets.

Network:	Regular	Issue 1	Issue 2	Issue 3	Issue 4	Issue 5
Original	99.49	100.0	99.80	93.09	99.99	94.94
Tailored	99.90	100.0	100.0	98.34	100.0	99.08

with missing values. To measure the performance, we report the Matthew Correlation Coefficient (MCC), normalised from -100 to 100. Whenever we evaluate our model, we repeat the entire process of generating the data, training, and testing 100 times and report the average result.

5. Improving the Rationale under Perfect Conditions

We first explore the rationale of our models under perfect conditions by training a neural network on the regular training dataset of 5,000 instances and then examining it using the TREI method. Following the first step of the method, we measure the performance of the trained network on the regular test dataset, which yields an average MCC of 99.49.

In the second step, we generate *rationale evaluation test sets* to evaluate how sound the decision-making of the neural network is. Recall that the verdict is defined as a conjunction of the five issues. The network must therefore have learned each issue in order for its decision making to be sound. To investigate to what extent the networks learned these individual issues, we generate rationale evaluation test sets.

For each Issue I_x, we create a different rationale evaluation test set, where all issues evaluate to true, except for Issue I_x. The base level factors (features) of Issue I_x are given random values (true or false). This way, the verdict is determined solely by whether Issue I_x evaluates to true, as the verdict is a conjunction of all issues. A network is therefore only able to predict the verdict of these specific rationale evaluation cases correctly if it has correctly learned Issue I_x. We create five such rationale evaluation test sets, one to test for each of the five issues. Each rationale evaluation test set contains 5,000 instances.

In step three, we task the trained neural network with classifying the rationale evaluation datasets and measure its performance in terms of MCC. The results can be seen in Table 1 (top row), alongside the MCC of the network on the regular test set.

Compared to the classification MCC of 99.49 on the regular dataset, some of the issues were not learned as well by the network. This is most likely due to the differences in complexity of the issues (see [5] for the full ADF). Issue 1, which is a simple 1-variable Boolean function, was picked up by the system, as it has a 100% accuracy on its respective rationale evaluation dataset. Issues 2 and 4 were also quite simple: both 3-variable conjunctions, which the network was able to learn with an MCC of 99.80 and 99.96 respectively. Issues 3 and 5, which are the more complex issues of the ADF, were not as learned as well, with an MCC of 93.09 and 94.94 respectively.

That some conditions were better learned than others can also be seen in in Figure 1, where we apply SHAP [10], an explainable AI method, to calculate the impact of each base level factor in the classification process of our network (blue bars). In this figure, we simplified the names of the base level factors, such that the xth factor of the yth issue is represented as $IyFx$. While explainable methods that extrapolate feature importance cannot guarantee us a sound rationale [11], they can tell us what base level factors were used by the network. The blue bars in the figure show that all base level factors are used

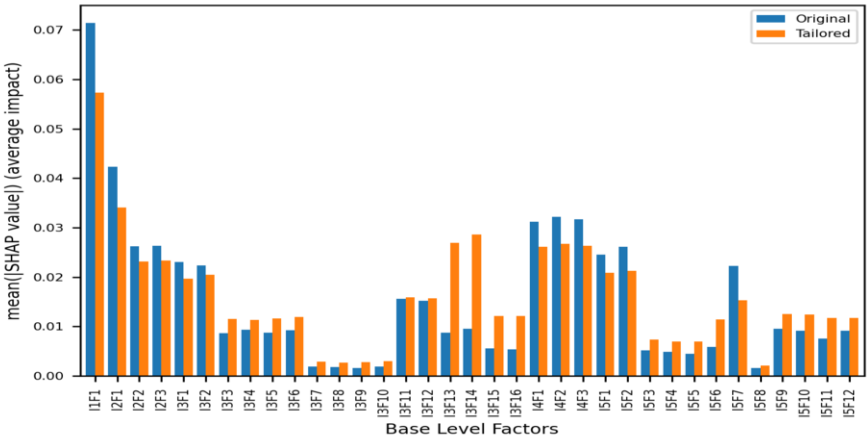

Figure 1. Mean SHAP values of the original neural network (blue) and the tailored neural network (orange) on the regular and rationale evaluation test sets for each feature (base level factor) in the dataset.

by the network for determining the verdict, but some have a higher impact than others, especially the base level factors belonging to issues I_3 and I_5, which confirms the rational evaluation results.

To improve the rationale of the network, we generate a *tailored training dataset* as described in step 4 of the TREI method. Based on the evaluation of the rationale, we know that a few issues were not learned as well by the networks. In the original training data, *at least* one of the issues would be false if the case did not violate Article 6. Statistically, this means that multiple issues usually evaluate to false in those cases. Providing the network with multiple issues at the same time might be too complex, and thus prompt the network to learn spurious correlations. In this new tailored training dataset, we therefore generate cases in which only one issue evaluates to false when the case evaluates to a non-violation.

The tailored dataset also consists of 5,000 cases, where half of the cases violate Article 6, and the other half do not. However, in the cases that do not violate Article 6, only one issue is false. This should make it easier for the network to pick up on the individual issues. Following step 4, we repeat the TREI method using this new tailored dataset. The same regular test set and rationale evaluation sets are used as before. The results can be seen in Table 1 (bottom row).

While the original network performed well on the regular test set, the network trained on the tailored training dataset performs slightly better. When it comes to making the correct decisions, it is therefore better to train on the tailored dataset. This tailored version of the network also performs better on all of the rationale evaluation datasets, indicating that the rationale of the network has improved. This can also be seen in Figure 1, where we plot the SHAP values of the tailored network (orange bars). While still giving a high impact value to simpler conditions, such as Issue 1, 2 or 4, which all reach an MCC of 100% on the rationale evaluation test sets (see Table 1), the tailored network also gives higher impact values than the original network to the factors belonging to Issue 3 and 5, which also have an increase in MCC on the rationale evaluation test sets.

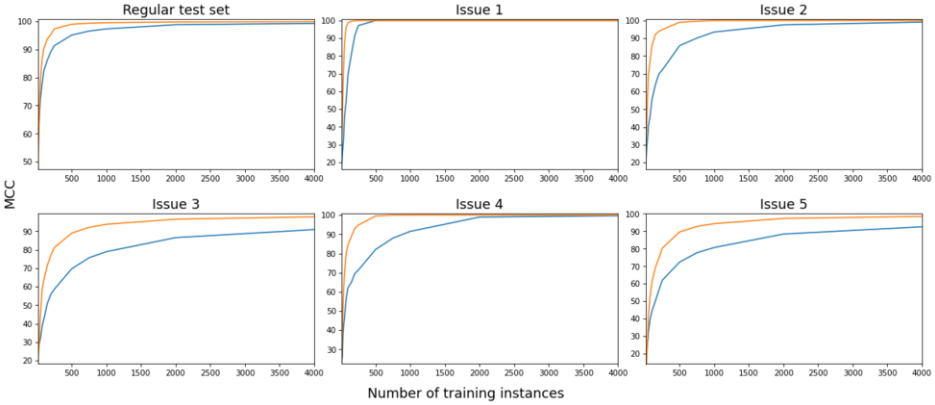

Figure 2. Mean MCC values of the original neural network (blue) and the tailored neural network (orange) on the regular and rationale evaluation test sets for varying amounts of training data.

6. Improving the Rationale under Imperfect Conditions

Under ideal circumstances, the rationale of our network was evaluated and improved by tailoring the training dataset. We now investigate the rationale improvement under imperfect conditions: small dataset sizes, inconsistencies or missing values. We only investigate one type of imperfection at a time. We train networks on both original type datasets and tailored type datasets. We refer to these networks as the original network and tailored network respectively. For each network there are therefore two independent variables: the type of training data (original or tailored) and the level of imperfection (training dataset size, inconsistency or missing values). While the networks are trained on imperfect training datasets in these experiments, they are tested on the same type of regular test sets and rationale evaluation test sets as in the previous section: without imperfections.

6.1. Training Dataset Size

To investigate the effects of training dataset size, we train networks on original and tailored datasets, ranging from 10 to 40,000 instances and then apply the TREI method. These datasets contain no inconsistencies or missing values. The mean MCC values across 100 runs can be seen in the plots of Figure 2. Note that each y-axis is scaled dynamically.

Discussion Two general remarks can be made from observing the plots in Figure 2. First of all, the original network and the tailored network perform better with more training data on all test sets. As we expected, the performance on the regular test set increases, but the performance on the rationale evaluation test sets increases as well. This suggests that, in this case, more data leads not only to a higher performance, but also to a better rationale. Secondly, the network trained on the tailored dataset outperforms the network trained on the original dataset. This is also true for both the performance on the regular dataset and the performance on the rationale evaluation test sets. The difference in MCC between the original and tailored network is highest with small training dataset sizes, such as the ones typically used in AI & Law research. This difference does decrease with

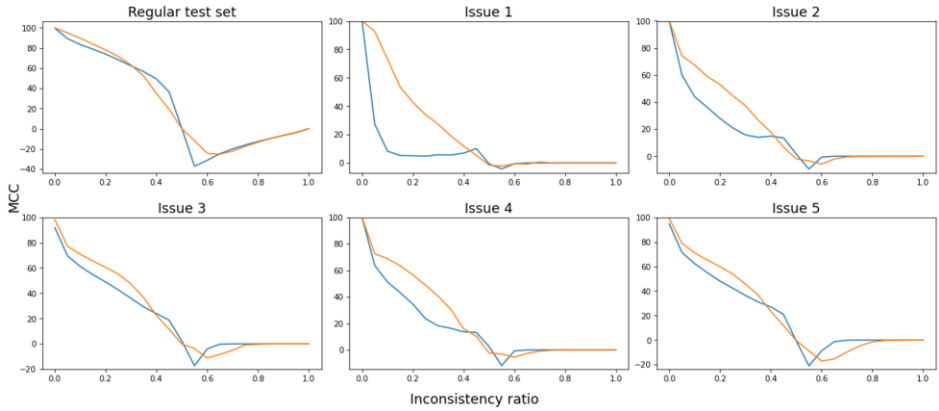

Figure 3. Mean MCC values of the original neural network (blue) and the tailored neural network (orange) on the regular and rationale evaluation test sets for varying amounts of inconsistencies in the training data.

more training data. This can be attributed to a ceiling effect, as the networks approach 100% accuracy. Despite the ceiling effect, however, the tailored network still performs better.

6.2. Inconsistencies

To investigate the effects of inconsistency, we generate original and tailored training datasets of varying levels of inconsistency (from 0% to 100% in steps of 5%), each containing 5,000 instances. We train networks on these datasets and apply the TREI method. The results across 100 runs can be seen in Figure 3. Note that each y-axis is scaled dynamically.

Discussion We can see in Figure 3, that more inconsistency leads to a lower performance (MCC on the test dataset) but also to a worse rationale (MCC on the rationale evaluation test sets). At a low inconsistency level, below 30%, the tailored network performs better on the regular test set than the original network. The tailored network also tends to outperform the original network with less than around 40% inconsistency on all of the rationale evaluation test sets. With more inconsistencies, the performances of the two networks drop and reach 0 or negative MCC values, indicating a performance that is worse than random chance. There is a sharp dip in all graphs around 50% inconsistency, after which the graph slowly rises again. This is when half of all features of all cases are flipped. After that, the majority of features is flipped, prompting the network to learn an inverted version of the ADF as discussed in Section 3, rather than the actual ADF, yielding an MCC of around 0. In reality, datasets where more than 30% of the facts are incorrect are not common. This would entail that, on average, all cases contain around 10 flipped factors. Therefore, using the tailored dataset yields a better performance and rationale regardless of the amount of inconsistency in the training data.

6.3. Missing Values

We generate original and tailored training datasets of 5,000 instances with varying levels of missing values (from 0% to 100% in steps of 5%) to train the networks. The results of

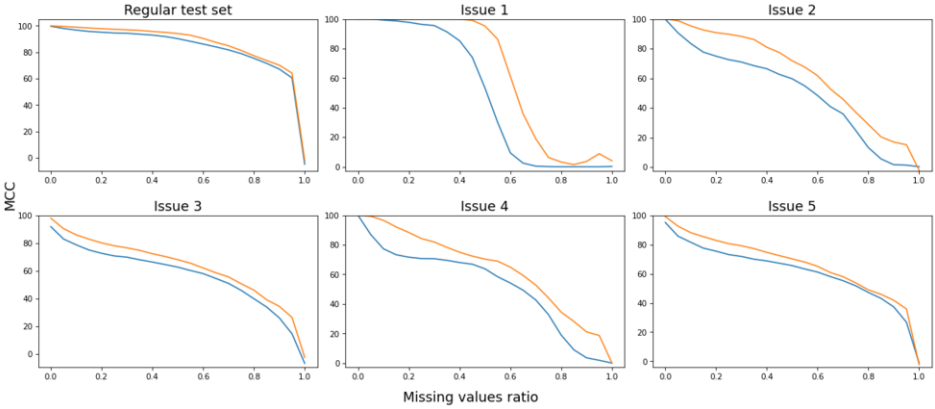

Figure 4. Mean MCC values of the original neural network (blue) and the tailored neural network (orange) on the regular and rationale evaluation test sets for varying amounts of missing facts in the training data.

applying the TREI method to these networks across 100 runs can be found in Figure 4. Note that each y-axis is scaled dynamically.

Discussion In Figure 4 we see a decrease in MCC on the regular test set with training datasets with more missing values. The rationale also seems to worsen, as evident from the MCC on the rationale evaluation test sets. With more missing values, we eventually see the MCC drop to 0 on all test cases. At a missing value rate of 1.0, all features of all cases are set to 0.5, meaning that the network has no way of distinguishing between violation or non-violation. Up until that point, the tailored network outperforms the original network for all tested levels of missing values.

7. Discussion and Conclusion

We generate artificial cases pertaining to Article 6 of the ECHR based on an abstract dialectical framework [5], which we use to train and test neural networks. By applying the TREI method to these networks, we are able to investigate where their rationale goes wrong, and based on that evaluation, we can create tailored training datasets that can be used to train networks that yield both a higher performance and a better rationale. This holds true in all of our experiments, even under imperfect conditions that that are common in AI & Law: small dataset sizes, inconsistencies and missing values.

In our experiments we focused on a controlled example, where full domain knowledge was available. In these type of scenarios, a structured knowledge representation, such as the ADF from which the data was generated, would outperform the neural networks that we used in terms of performance, rationale and explainability. In this study, we show that the rationale of black box machine learning systems, simple neural networks, can be evaluated and improved using the TREI method under imperfect conditions. The three imperfect conditions that we study are not an exhaustive list of issues, and we acknowledge that there are more concerns in AI & Law than just small dataset sizes, inconsistencies and missing values. The methods for introducing inconsistency and missing values in the datasets may be varied as well. A different, more realistic, approach to creating inconsistency might be to flip the labels rather than the features, or

to flip clustered groups of related features. Similarly, missing values should in practice only be absent from irrelevant aspects, which we did not account for in this study. The current implementation serves to illustrate how rationales can still be improved using the TREI method under inconsistency and missing values. We leave the variations for future research.

We have expanded upon previous research, where the TREI method was tested under perfect conditions, by investigating rationale improvement under imperfect conditions. In earlier research, improving the rationale on the fictional welfare benefit domain caused a slight decrease in the overall performance of the models [3]. Improving the rationale in this study also improved the general performance. Additionally, while the first step of the method states to only proceed once the performance is sufficiently high, we have shown that improvement in performance and rationale is possible even with poor performing models. Our results show that the TREI method is not only useful under perfect conditions, but also under imperfect conditions within AI & Law.

Acknowledgments

This research was funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl.

References

[1] Bench-Capon T. Neural Networks and Open Texture. In: Proceedings of the 4th International Conference on Artificial Intelligence and Law. ICAIL '93. ACM, New York; 1993. p. 292-7.
[2] Barredo Arrieta A, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion. 2020;58:82-115.
[3] Steging C, Renooij S, Verheij B. Discovering the rationale of decisions: towards a method for aligning learning and reasoning. In: Maranhão J, Wyner AZ, editors. ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021. ACM; 2021. p. 235-9.
[4] Bench-Capon T. The need for good old fashioned AI and law. International trends in legal informatics: a Festschrift for Erich Schweighofer. 2020:23-36.
[5] Collenette J, Atkinson K, Bench-Capon TJM. Explainable AI tools for legal reasoning about cases: A study on the European Court of Human Rights. Artif Intell. 2023;317:103861.
[6] Medvedeva M, Wieling M, Vols M. Rethinking the field of automatic prediction of court decisions. Artificial Intelligence and Law. 2022:1-18.
[7] Atkinson K, Bench-Capon T. ANGELIC II: An Improved Methodology for Representing Legal Domain Knowledge. 2023.
[8] Mumford J, Atkinson K, Bench-Capon T. Reasoning with Legal Cases: A Hybrid ADF-ML Approach. Legal Knowledge and Information Systems. 2022;362:93-102.
[9] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825-30.
[10] Lundberg SM, Lee S. A unified approach to interpreting model predictions. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems 30. Curran Associates, Inc.; 2017. p. 4765-74.
[11] Steging C, Renooij S, Verheij B. Rationale Discovery and Explainable AI. In: Schweighofer E, editor. Legal Knowledge and Information Systems - JURIX 2021: The Thirty-fourth Annual Conference, Vilnius, Lithuania, 8-10 December 2021. vol. 346 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2021. p. 225-34.