Artificial Intelligence 296 (2021) 103471

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Using ontologies to enhance human understandability of global post-hoc explanations of black-box models ☆

Roberto Confalonieri [a,*], Tillman Weyde [b], Tarek R. Besold [c], Fermín Moscoso del Prado Martín [d]

[a] Free University of Bozen-Bolzano, Faculty of Computer Science, I-39100 Bozen-Bolzano, Italy
[b] Dept. of Computer Science, City University of London, GB-EC1V 0HB London, United Kingdom
[c] Neurocat GmbH, Rudower Chaussee 29, D-12489 Berlin, Germany
[d] Lingvist Technologies OÜ, Tallinn, Estonia

ARTICLE INFO

Article history:
Received 23 April 2020
Received in revised form 24 December 2020
Accepted 8 February 2021
Available online 15 February 2021

Keywords:
Human-understandable explainable AI
Global explanations
Ontologies
Neural-symbolic learning and reasoning
Knowledge extraction
Concept refinement

ABSTRACT

The interest in explainable artificial intelligence has grown strongly in recent years because of the need to convey safety and trust in the 'how' and 'why' of automated decision-making to users. While a plethora of approaches has been developed, only a few focus on how to use domain knowledge and how this influences the understanding of explanations by users. In this paper, we show that by using ontologies we can improve the human understandability of global post-hoc explanations, presented in the form of decision trees. In particular, we introduce Trepan Reloaded, which builds on Trepan, an algorithm that extracts surrogate decision trees from black-box models. Trepan Reloaded includes ontologies, that model domain knowledge, in the process of extracting explanations to improve their understandability. We tested the understandability of the extracted explanations by humans in a user study with four different tasks. We evaluate the results in terms of response times and correctness, subjective ease of understanding and confidence, and similarity of free text responses. The results show that decision trees generated with Trepan Reloaded, taking into account domain knowledge, are significantly more understandable throughout than those generated by standard Trepan. The enhanced understandability of post-hoc explanations is achieved with little compromise on the accuracy with which the surrogate decision trees replicate the behaviour of the original neural network models.

1. Introduction

In recent years, explainability has been identified as a key factor for the adoption of AI systems in a wide range of contexts [11,22,29,39,43,55]. The emergence of intelligent systems in self-driving cars, medical diagnosis, insurance and financial services among others has shown that is essential to provide an explanation to users, developers or regulators when decisions are taken or suggested by automated systems for practical, social, and increasingly also legal reasons. As a case in

☆ This paper is part of the Special Issue on Explainable AI.

* Corresponding author.
E-mail address: roberto.confalonieri@unibz.it (R. Confalonieri).

https://doi.org/10.1016/j.artint.2021.103471

point, the European Union's General Data Protection Regulation (GDPR) stipulates a right to "*meaningful information about the logic involved*"—commonly interpreted as a 'right to an explanation'—for consumers affected by an automatic decision [48].[1]

User rights and acceptance are not the only reasons for equipping intelligent systems with explanation capabilities. Explainability is also needed for designers and developers to enhance system robustness [5] and enable diagnostics to prevent bias, unfairness and discrimination [41,27], as well as for increasing trust by all users in *why* and *how* decisions are made [55]. Against that background, increasing efforts are directed towards studying and provisioning explainable intelligent systems, both in industry and academia, sparked by initiatives like the DARPA Explainable Artificial Intelligence Program (XAI) [18] and leading a growing number of scientific conferences and workshops dedicated to explainability that are now regularly organised (e.g., the 'ACM Conference on Fairness, Accountability, and Transparency' – ACM FAccT, or the 'Workshop on Explainable Artificial Intelligence' at several editions of the International Joint Conference on Artificial Intelligence).

While interest in XAI had subsided together with that in expert systems after the mid-1980s [8,67], more recent successes in machine learning technology have brought explainability back into the focus [15]. This has led to a plethora of new approaches for *post-hoc* explanations of black-box models [26], aiming to achieve explainability without sacrificing system performance. These are used for both autonomous and human-in-the-loop systems [30,31], applied to areas such as recommender systems [46,62] and using methods such as neural-symbolic reasoning and learning [25]. Only a few of these approaches, however, focus on global explanations, and on how to integrate and use domain knowledge to drive the explanation process (e.g., [47,53]) or how to measure the understandability of explanations of black-box models (e.g., [56]). For that reason, an important foundational aspect of explainable AI has remained hitherto mostly unexplored: can the integration of domain knowledge, e.g., as modelled by means of ontologies, help human understandability of explanations?

To tackle this research question, we propose a neural-symbolic learning approach based on Trepan [17], an algorithm devised to explain trained artificial neural networks by means of decision trees, which we extend to take into account ontologies in the tree extraction process. In particular, we modify the creation of split nodes to prefer features associated with more general concepts in a domain ontology. Linking explanations to structured knowledge, in the form of ontologies, brings multiple advantages. It does not only enrich explanations (or the elements therein) with semantic information—thus facilitating effective knowledge transmission to users—, but it also creates a potential for supporting the customisation of the levels of specificity and generality of explanations to specific user profiles [28].

We further designed and conducted an on-line user study to measure the effects of the ontology on the understandability of explanations with human users in the domains of finance and medicine, where explanations are critical. Our study shows that decision trees generated by Trepan Reloaded, thus taking domain knowledge into account, are more understandable than those generated without the use of domain knowledge. This augmented human understandability is measured through time and accuracy of responses as well as reported user confidence and understandability, and similarity of free text responses. Crucially, the enhanced understandability of the resulting trees is achieved with little compromise on the accuracy with which the resulting trees replicate the behaviour of the original neural network model.[2]

In summary, the contribution of this paper is twofold:

- We propose Trepan Reloaded, a model-agnostic algorithm for explaining classifiers using semantic information provided in an ontology. The algorithm gives preference to input features that are associated to more abstract concepts.
- We evaluate Trepan Reloaded in a user experiment. We find that its use of semantic information robustly increases the human understanding of global explanations for black-box classifiers.

The remainder of the paper is organised as follows. After introducing Trepan, and the notion of ontologies in Section 2, we present our extended algorithm, Trepan Reloaded, that uses ontologies in the decision tree extraction in Section 3. In Section 4, we propose how to measure understandability of decision trees from a technical and a user perspective. Section 5 reports and analyses the results of our experiment. After discussing our approach in Section 6, Section 7 situates our results in the context of related research. Finally, Section 8 concludes the paper and outlines possible future work.

2. Preliminaries

In this section, we present the foundations of our approach, namely, decision trees, the Trepan algorithm and formal ontologies.

2.1. Decision trees

Decision trees are one of the most popular machine learning models for classification and regression problems. They are popular since they are symbolic models that are intrinsically interpretable in the sense that the user can follow the path through the tree and trace the decision process. For this reason, decision trees are often used as local or global explanations of more opaque learning models such as neural networks or support vector machines [26].

[1] Regulation (EU) 2016/679 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) [2016] OJ L119/1.

[2] The dataset and the analysis performed is published for reproducibility in [13].

A decision tree is a rooted, directed acyclic graph consisting of a set of split nodes, usually depicted as rectangles, and a set of leaves, usually depicted as ovals. Each split node in a decision tree has an associated logical test based on the features in the domain. When classifying an instance or example, the role of a split node is to assign the example to one of the outgoing branches of the node. Fig. 2 shows an example of a decision tree.

Split nodes may have several branches depending on whether the logical test is over binary, nominal, or real values attributes. The decision as to which branch is selected for an example is determined by the logical test of the node. In the simplest case, this test considers one feature, and thus the outcome of the test is determined by the value of that feature in the given example.

The way in which split nodes are selected amounts to optimising a reward function, the definition of which can vary depending on the type of induction algorithm used. In Trepan, the split selection is generally based on *information gain*. Given a set of features $X = \{X_1, \ldots, X_n\}$, the information gain IG of a feature X_i w.r.t. a set of samples S is defined as

$$\mathsf{IG}(X_i, S) := H(S) - \sum_{j=1}^{k} \frac{|S_j|}{|S|} H(S_j) \ . \tag{1}$$

where S_j is the subset of samples in S that have value j of feature X_i and $H(S)$ is the entropy of S, which is defined as

$$H(S) := -\sum_{c \in C} \frac{|S^c|}{|S|} \log_2 \frac{|S^c|}{|S|}.$$

where S_c is the subset of example in S that belong to a class c. The performance of decision trees for classification tasks is measured by *accuracy*, i.e., the fraction of predictions that agree with the ground truth.

In this paper, our objective is to improve the understandability of decision trees extracted by Trepan. To this end, we will extend the reward function of a feature X_i to take into account its role in an ontology, as detailed in Section 3.

2.2. *The* Trepan *algorithm*

Trepan is a tree induction algorithm that recursively extracts decision trees from oracles, in particular from feed-forward neural networks [17]. The original motivation behind the development of Trepan was to approximate a neural network by means of a symbolic structure that is more interpretable than a neural network classification model. This was in the context of a wider interest in knowledge extraction from neural networks (see [63,19] for an overview).

Algorithm 1 Trepan(*Oracle*,*Training*,*Features*).

```
Priority queue Q ← ∅
Tree T ← ∅
use Oracle to label examples in Training
enqueue root node into Q
while nr_internal_nodes < size_limit do
    pop node n from Q
    draw and store examples_n for n
    store constraints_n for n
    use features to build set of candidate splits
    use examples_n and Oracle(constraints_n)
        to decide Best_split
    add n to T
    for element c ∈ Best_split do
        add c as child of n
        if c is not a leaf according to Oracle(constraints_n) then
            enqueue node c into Q with negative
                information gain as priority
        end if
    end for
end while
Return T
```

The pseudo-code for Trepan is shown in Algorithm 1. Trepan differs from conventional inductive learning algorithms as it uses an *oracle* to classify examples during the learning process. It generates new examples by sampling from distributions over the given examples and constraints (conditions that examples must satisfy in order to reach a node), so that the amount of training data used to select splitting tests and to label leaves does not decrease with the depth of the tree. It expands a tree in a best-first manner by means of a priority queue by entropy, that prioritises nodes that have greater potential for improvement. Further details of the algorithm can be found in [16].

$\mathsf{Entity} \sqsubseteq \top$	,	$\mathsf{Person} \sqsubseteq \mathsf{PhysicalObject}$
$\mathsf{AbstractObject} \sqsubseteq \mathsf{Entity}$	,	$\mathsf{Loan} \sqsubseteq \mathsf{AbstractObject}$
$\mathsf{PhysicalObject} \sqsubseteq \mathsf{Entity}$	,	$\mathsf{Gender} \sqsubseteq \mathsf{Quality}$
$\mathsf{Quality} \sqsubseteq \mathsf{Entity}$	,	$\mathsf{Male} \sqsubseteq \mathsf{Gender}$
$\mathsf{LoanApplicant} \sqsubseteq \mathsf{Person} \sqcap \exists \mathsf{hasApplied.Loan}$	,	$\mathsf{Female} \sqsubseteq \mathsf{Gender}$

Fig. 1. An ontology excerpt for the loan domain. The ontology is formalised in the $\mathcal{EL}$ Description Logic, but the techniques introduced in the paper apply to a variety of logics.

Trepan stops the tree extraction process when one of two criteria is met: no more nodes need to be further expanded because their entropy is low enough (they contain almost exclusively instances of a single class) or a predefined limit of the tree size (the number of nodes) is reached. While Trepan was designed to explain neural networks as the oracle, it is model-agnostic and can be used to explain any other classification model.

In order to evaluate the performance of algorithms like Trepan that approximate an initial machine learning model with a so-called surrogate, we measure the accuracy and the fidelity of the resulting surrogate, i.e. the decision trees. *Fidelity* is defined as the percentage of examples on which the classification by a surrogate agrees with that by the initial model. Fidelity is more relevant here than accuracy, which compares to the ground truth, as it is the direct measure the goal of the surrogate model.

2.3. Ontologies

An ontology is a set of formulae in an appropriate logical language with the purpose of describing a particular domain of interests. The specific logic used is not crucial for our approach as the techniques introduced here apply to a variety of logics. For the sake of clarity we use description logics (DLs) as well-known ontology languages. We briefly introduce the DL $\mathcal{EL}$, a DL allowing only conjunctions and existential restrictions. For full details, see [2].

Syntactically, $\mathcal{EL}$ is based on two disjoint sets N_C and N_R of *concept names* and *role names*, respectively. The set of $\mathcal{EL}$ *concepts* is generated by the grammar

$$C ::= A \mid C \sqcap C \mid \exists R.C,$$

where $A \in N_C$ and $R \in N_R$. A *TBox* is a finite set of general concept inclusions (GCIs) of the form $C \sqsubseteq D$ where C and D are concepts. It stores the terminological knowledge describing the relationships between concepts. An *ABox* is a finite set of assertions $C(a)$ and $R(a, b)$, which express knowledge about objects in the knowledge domain. An *ontology* is composed by a TBox and an ABox. In this paper, we focus on the TBox only, thus we will use the terms ontology and TBox interchangeably.

The semantics of $\mathcal{EL}$ is based on *interpretations* of the form $I = (\Delta^I, \cdot^I)$, where Δ^I is a non-empty *domain*, and $\cdot^I$ is a function mapping every individual name to an element of Δ^I, each concept name to a subset of the domain, and each role name to a binary relation on the domain. I satisfies $C \sqsubseteq D$ iff $C^I \subseteq D^I$ and I satisfies an assertion $C(a)$ ($R(a, b)$) iff $a^I \in C^I$ ($(a^I, b^I) \in R^I$). The interpretation $\mathcal{I}$ is a *model* of the TBox $\mathcal{T}$ if it satisfies all the GCIs and all the assertions in $\mathcal{T}$. $\mathcal{T}$ is *consistent* if it has a model. Given two concepts C and D, C is *subsumed* by D w.r.t. $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) if $C^I \subseteq D^I$ for every model I of $\mathcal{T}$. We write $C \equiv_{\mathcal{T}} D$ when $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$. C is *strictly subsumed by* D w.r.t. $\mathcal{T}$ ($C \sqsubset_{\mathcal{T}} D$) if $C \sqsubseteq_{\mathcal{T}} D$ and $C \not\equiv_{\mathcal{T}} D$. We denote by $\mathcal{L}(\mathcal{EL}, N_C, N_R)$ the set of (complex) concepts built over N_C and N_R in $\mathcal{EL}$.

Fig. 1 shows an ontology excerpt modelling concepts and relations relevant to the *loan* domain. The precise formalisation of the domain is not relevant at this point; different formalisations may exist, with different levels of granularity. The ontology structures the domain knowledge from the most *general* concept (e.g., Entity) to more *specific* concepts (e.g., LoanApplicant, Female, etc.). The subsumption relation ($\sqsubseteq$) induces a partial order among the concepts that can be built from a TBox $\mathcal{T}$. For instance, the Quality concept is more general than the Gender concept, and it is more specific than the Entity concept.

We will capture the degree of generality (resp. specificity) of a concept in terms of an information content measure that is based on concept refinement. The measure is defined in detail in Section 3 and serves as the basis for the subsequent extension of the Trepan algorithm.

2.4. Concept refinement

The idea behind concept refinement is to make a concept more general or more specific by means of refinement operators. Refinement operators are well-known in Inductive Logic Programming, where they are used to learn concepts from examples. In this setting, two types of refinement operators exist: specialisation refinement operators and generalisation refinement operators. While the former construct specialisations of hypotheses, the latter construct generalisations [65]. In this paper, we focus on specialisation operators.

Given the quasi-ordered set $\langle \mathcal{L}(\mathcal{EL}, N_c, N_R), \sqsubseteq \rangle$, a specialisation refinement operator satisfies:

$$\rho_{\mathcal{T}}(C) \subseteq \{C' \in \mathcal{L}(\mathcal{EL}, N_c, N_R) \mid C' \sqsubseteq_{\mathcal{T}} C\}.$$

Specialisation refinement operators take a concept C as input and return a set of descriptions that are more specific than C by taking a TBox $\mathcal{T}$ into account. Refinement operators for description logics were introduced in [37], and further developed in [10,64]. Several definitions of refinement operators exist but our approach is independent of their specifics. The only requirement is that the operator is finite, that is, the set of refinements computed by the operator is finite. When a specific refinement operator is needed, as in the examples and in the experiments, we use the specialisation operator defined in what follows.

A refinement operator is defined by three things: i) the (finite) set of subconcepts that can be formed from a TBox $\mathcal{T}$; ii) the upward cover set of a concept C, and iii) a set of transformation rules defined over the structure of concept descriptions.

The finite set of subconcepts that can be built from the axioms of a TBox $\mathcal{T}$ is obtained by structural induction over the concept descriptions. This set is denoted by $\mathsf{sub}(\mathcal{T})$ and is defined as follows.

Definition 2.1. Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox. The set of *subconcepts* of $\mathcal{T}$ is given as

$$\mathsf{sub}(\mathcal{T}) := \{\top, \bot\} \cup \bigcup_{C \sqsubseteq D \in \mathcal{T}} \mathsf{sub}(C) \cup \mathsf{sub}(D),$$

where sub is inductively defined over the structure of concept descriptions as follows:

$$\begin{aligned}
\mathsf{sub}(A) &:= \{A\} \\
\mathsf{sub}(\top) &:= \{\top\} \\
\mathsf{sub}(C \sqcap D) &:= \{C \sqcap D\} \cup \mathsf{sub}(C) \cup \mathsf{sub}(D) \\
\mathsf{sub}(\exists r.C) &:= \{\exists r.C\} \cup \mathsf{sub}(C)
\end{aligned}$$

We also denote the set of subconcepts of a concept C by $\mathsf{subConcepts}(C)$.

Based on $\mathsf{sub}(\mathcal{T})$, we define the downcover set of atomic concepts and roles. $\mathsf{sub}(\mathcal{T})$ guarantees the following downcover set to be finite.[3] The downcover sets of a concept C contain the most general subconcepts and roles found in $\mathsf{sub}(\mathcal{T})$ that are more specific than (are subsumed by) C.

Definition 2.2. Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox, C a concept. The *downcover* sets of C w.r.t. $\mathcal{T}$ is:

$$\mathsf{DownCov}_{\mathcal{T}}(C) := \{D \in \mathsf{sub}(\mathcal{T}) \mid D \sqsubseteq_{\mathcal{T}} C \text{ and } \nexists D'. D' \in \mathsf{sub}(\mathcal{T}) \text{ with } D \sqsubset_{\mathcal{T}} D' \sqsubset_{\mathcal{T}} C\}.$$

Given the previous definitions a specialisation refinement operator for $\mathcal{EL}$ concepts can be defined as follows.

Definition 2.3. Given a Tbox $\mathcal{T}$ and a concept description C, a specialisation operator $\rho_{\mathcal{T}}(C)$ is defined as:

$$\begin{aligned}
\rho(A) &:= \mathsf{DownCov}_{\mathcal{T}}(A) \\
\rho(\top) &:= \mathsf{DownCov}_{\mathcal{T}}(\top) \\
\rho(\bot) &:= \mathsf{DownCov}_{\mathcal{T}}(\bot) \\
\rho(C \sqcap D) &:= \mathsf{DownCov}_{\mathcal{T}}(C \sqcap D) \\
\rho(\exists r.C) &:= \mathsf{DownCov}_{\mathcal{T}}(\exists r.C)
\end{aligned}$$

A concept C can specialised by any of its most general specialisations that belong to $\mathsf{sub}(\mathcal{T})$ using the unbounded finite iteration of ρ.

Definition 2.4. The unbounded finite iteration of the refinement operator ρ is defined as:

$$\rho^*_{\mathcal{T}}(C) := \bigcup_{i \geq 0} \rho^i_{\mathcal{T}}(C) \ .$$

where $\rho^i_{\mathcal{T}}(C)$ is inductively defined as:

$$\begin{aligned}
\rho^0_{\mathcal{T}}(C) &:= \{C\}, \\
\rho^{j+1}_{\mathcal{T}}(C) &:= \rho^j_{\mathcal{T}}(C) \cup \bigcup_{C' \in \rho^j_{\mathcal{T}}(C)} \rho_{\mathcal{T}}(C'), j \geq 0.
\end{aligned}$$

[3] We assume that $\mathcal{T}$ is finite.

$\rho^*_{\mathcal{T}}(C)$ is the set of subconcepts of C w.r.t. $\mathcal{T}$. We will denote this set by subConcept(C). Since sub($\mathcal{T}$) is a finite set, the operator $\rho^*_{\mathcal{T}}(C)$ is finite, and it terminates (i.e., $\bot \in \rho^*_{\mathcal{T}}(C)$). For a detailed analysis of properties of refinement operators in DLs we refer to [37,10].

Example 1. Let us consider the concepts Entity, and LoanApplicant defined in the ontology in Fig. 1. Then:
$\rho_{\mathcal{T}}(\mathsf{Entity}) \subseteq \{\mathsf{Entity}, \mathsf{AbstractObject}, \mathsf{PhysicalObject}, \mathsf{Quality}\}$;
$\rho^*_{\mathcal{T}}(\mathsf{Entity}) \subseteq \mathsf{sub}(\mathcal{T})\backslash\{\top\}$;
$\rho_{\mathcal{T}}(\mathsf{LoanApplicant}) = \rho^*_{\mathcal{T}}(\mathsf{LoanApplicant}) \subseteq \{\mathsf{LoanApplicant}, \bot\}$.

3. TREPAN Reloaded

Our aim is to create decision trees that are more understandable for humans by prioritising more understandable features in the tree generation process. Our hypothesis, which we will validate in this study, is that features are more understandable if they relate to more general concepts present in an ontology.

To measure the degree of semantic generality or specificity of a concept, we consider its *information content* [59] as typically adopted in computational linguistics [54]. Classical information theoretic approaches compute the information content of a concept as the inverse of its frequency in a corpus, so that infrequent terms are considered more informative and less general than frequent ones.

In ontologies, the information content can be computed either extrinsically from the concept occurrences (e.g., [54]), or intrinsically, according to the number of subsumed concepts modelled in the ontology. Here, we adopt the latter approach. We use this information content measure to prioritise features that are more general (thus presenting less information content), as our hypothesis is that the decision tree becomes more understandable when it uses more general concepts.

Definition 3.1. Given an ontology $\mathcal{T}$, the information content of a feature X_i is defined as:

$$\mathsf{IC}(X_i) := \begin{cases} 1 - \dfrac{\log(|\mathsf{subConcepts}(X_i)|)}{\log(|\mathsf{sub}(\mathcal{T})|)} & \text{if } X_i \in \mathsf{sub}(\mathcal{T}) \\ 0 & \text{otherwise.} \end{cases}$$

where subConcepts(X_i) is the set of specialisations for X_i, and sub($\mathcal{T}$) is the set of subconcepts that can be built from the axioms in the TBox $\mathcal{T}$ of the ontology (see Section 2.4). It can readily be seen that the values of IC are smaller for features associated with more general and greater for features associated with more specific concepts.

Example 2. Let us consider the concepts Entity, and LoanApplicant defined in the ontology in Fig. 1 and the refinements in Example 1.
The cardinality of sub($\mathcal{T}$) is 13.
The cardinality of subConcepts(Entity) and subConcepts(LoanApplicant) is 12 and 2 respectively.
Then: $\mathsf{IC}(\mathsf{Entity}) = 0.04$, and $\mathsf{IC}(\mathsf{LoanApplicant}) = 0.73$.

With this method to compute the information content of a feature X_i, we now propose to update the information gain used by TREPAN to give preference to features with a lower information content.

Definition 3.2. The information gain given the information content IC of a feature X_i is defined as:

$$\mathsf{IG}'(X_i, S|\mathsf{IC}) := \begin{cases} (1 - \mathsf{IC}(X_i))\mathsf{IG}(X_i, S) & \text{if } 0 < \mathsf{IC}(X_i) < 1 \\ 0 & \text{otherwise.} \end{cases}$$

where $\mathsf{IG}(X_i, S)$ is the information gain as defined in Eq. (1).

IG′ of a feature is reduced compared to IG by a proportion that varies depending on its information content, and it is set to 0 either when the feature is not present in the ontology or when its information content is maximal. Although our definition of IG′ is specific to IG, analogously formulations are possible with other reward functions, such as the Gini impurity [7, Ch. 11].

Our hypothesis that using features associated with more general concepts in the creation of split nodes can enhance the understandability of the tree, is based on users being more familiar with more general concepts rather than more specialised ones. From a cognitive perspective this is also plausible, since more general concepts have been found to be easier to understand and learn [23]. To validate this hypothesis we ran a survey-based online study with human participants. Before proceeding to the details of the study, we introduce two measures for the understandability of a decision tree—an *objective*, syntax-based and a *subjective*, performance-based one—in the following section.

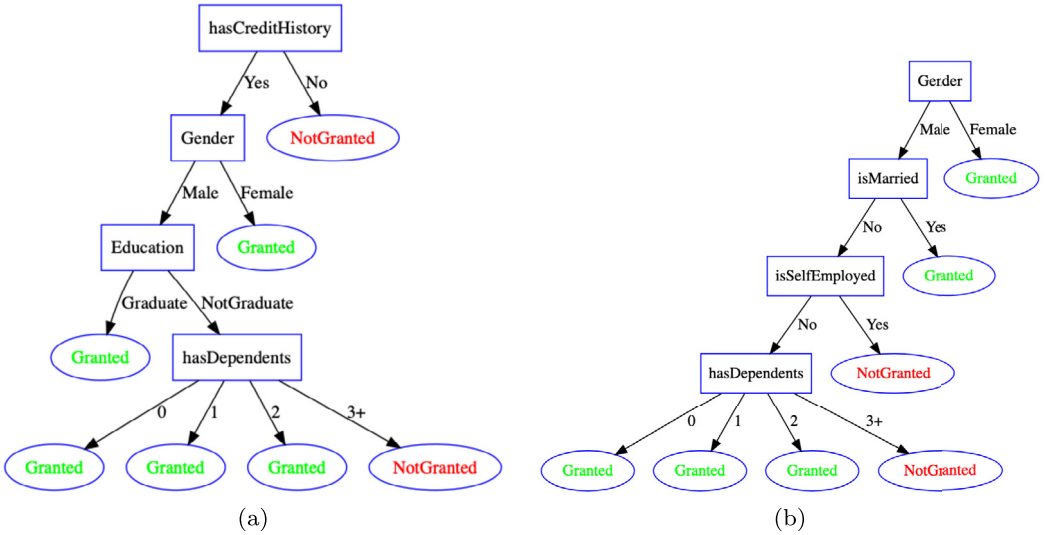

Fig. 2. Decision trees of size 'small' in the loan domain, extracted without (a) and with (b) a domain ontology. It can be seen that the use of an ontology leads to different features appearing in the split nodes. For instance, in the ontology used to build tree (b) the concept *Gender* is more abstract than *isMarried* and *isSelf Employed* has thus a lower information content according to Definition 3.1. Concepts with lower information content are favoured as conditions for split nodes in the tree according to Definition 3.2, which leads to *Gender* being used first by TREPAN-Reloaded when it generated the split nodes of tree (b). Furthermore, the ontology does not include concepts associated to *hasCreditHistory* and *Education*, which are therefore not considered in the construction of tree (b).

4. Understandability of decision trees

Understandability depends on the cognitive load experienced by users in using the decision model to classify instances and in understanding the features in the model itself. However, for practical processing human understandability of decision trees needs to be approximated by an objective measure. We compare here two characterisations of the understandability of decision trees:

- Understandability based on the syntactic complexity of a decision tree
- Understandability based on user performance and subjective ratings

Previous work attempting to measure the understandability of symbolic decision models (e.g., [32]), and decision trees in particular [51], proposed syntactic complexity measures based on the tree structure. The syntactic complexity of a decision tree can be measured, for instance, by counting the number of internal nodes, leaves, the number of symbols used in the splits (relevant especially for *m-of-n* splits), or the number of branches that decision nodes have.

For the sake of simplicity, we focus here on the combination of two syntactic measures: the number of leaves n in a decision tree, and the number of branches b on the paths from the root of the tree to all the leaves in the decision tree. Based on the results in [51], we define the *syntactic complexity* of a decision tree as:

$$U(n,b) := \alpha\frac{n}{k} + (1-\alpha)\frac{b}{k^2}\ . \tag{2}$$

with $\alpha \in [0,1]$ being a tuning factor that adjusts the weight of n and b, and $k = 5$ being the coefficient of the linear regression built using the results in [51].

Having a measure like syntactic complexity, that can be easily computed, is useful from an application perspective. E.g., it may be used to prevent excessive complexity in automatic tree generation.

On the other hand, the syntactic complexity of decision trees does not necessarily capture precisely the understandability for users. A direct measure of user understandability is how accurately a user can employ a given decision tree to perform a decision. Another measure of cognitive difficulty is the reaction time (RT) or response latency [21]. RT is a standard measure used by cognitive psychologists and has become a staple measure of complexity in the domain of design and user interfaces [68]. In our experiments we therefore measured the cost of processing in terms of accuracy, and RT, among other variables, for different types of decision trees.

The effects of tree generation and structure may be different for different tree sizes. For our experiments, we therefore defined three categories of tree sizes based on the number of internal nodes: *small* (between 0 and 10 internal nodes), *medium* (between 11 and 20 internal nodes), and *large* (between 21 and 30 internal nodes).

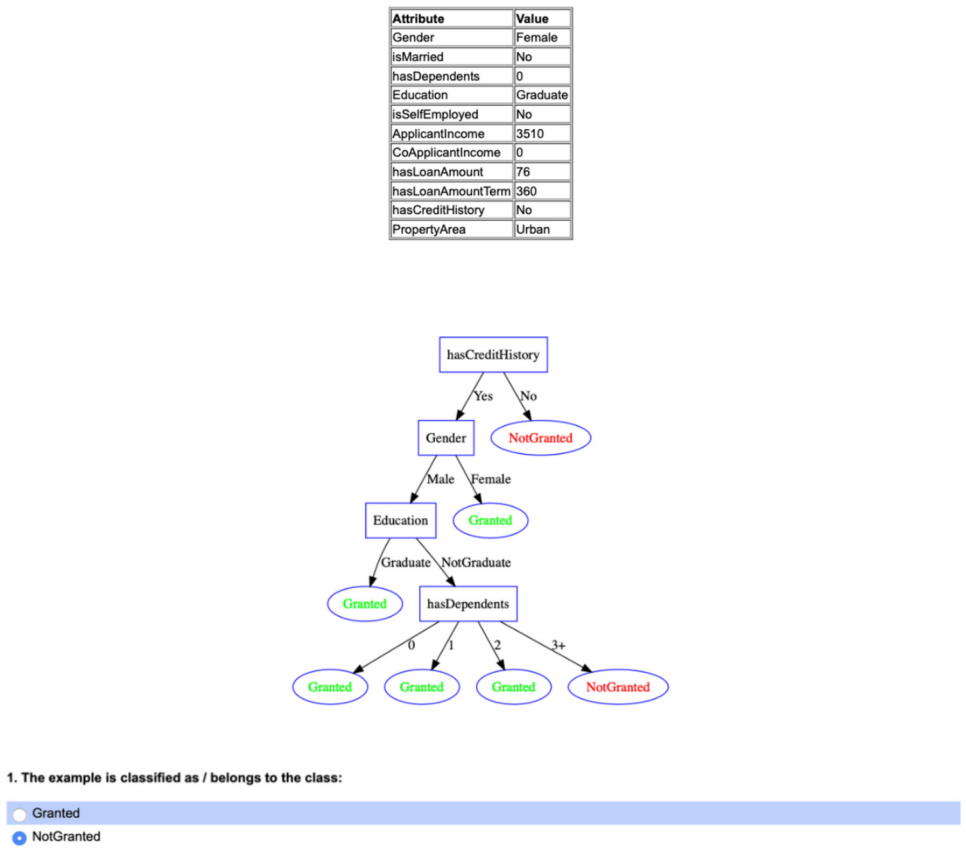

Fig. 3. Example of classification task. Respondents were asked to classify the example in the table using the decision tree (without knowing whether the tree was generated using an ontology or not). For each respondent, the reaction time and the accuracy of the answer were stored. After answering, respondents were also asked to rate their confidence in the answer and the subjective understandability of the decision tree.

5. Experimental evaluation

In our empirical evaluation, we ran user experiments on two datasets with a variety of tasks and different decision tree generation methods and sizes. We measured responses for evaluation of understandability.

5.1. Method

Materials. We used datasets from two different domains to evaluate our approach: finance and medicine. We used the *Cleveland Heart Disease Data Set* from the UCI archive,[4] and the *Loan Prediction Problem Dataset* from Kaggle.[5] For each dataset, we developed an ontology defining the main concepts and relevant relations (the heart and loan ontology contained 29 classes, 66 logical axioms and 28 classes, 65 logical axioms respectively). To extract decision trees using the TREPAN and TREPAN Reloaded algorithm, we trained two artificial neural networks implemented in *PyTorch*. The neural networks we use in our experiments have a single layer of hidden units. The number of hidden units in each network is chosen using cross-validation on the network's training set, and we use a validation set to determine when to stop training networks. The test set accuracy achieved with the trained neural networks was of 85.98% and 94.65% for the loan and heart dataset respec-

[4] http://archive.ics.uci.edu/ml/datasets/Heart+Disease.
[5] https://www.kaggle.com/altruistdelhite04/loan-prediction-problem-dataset.

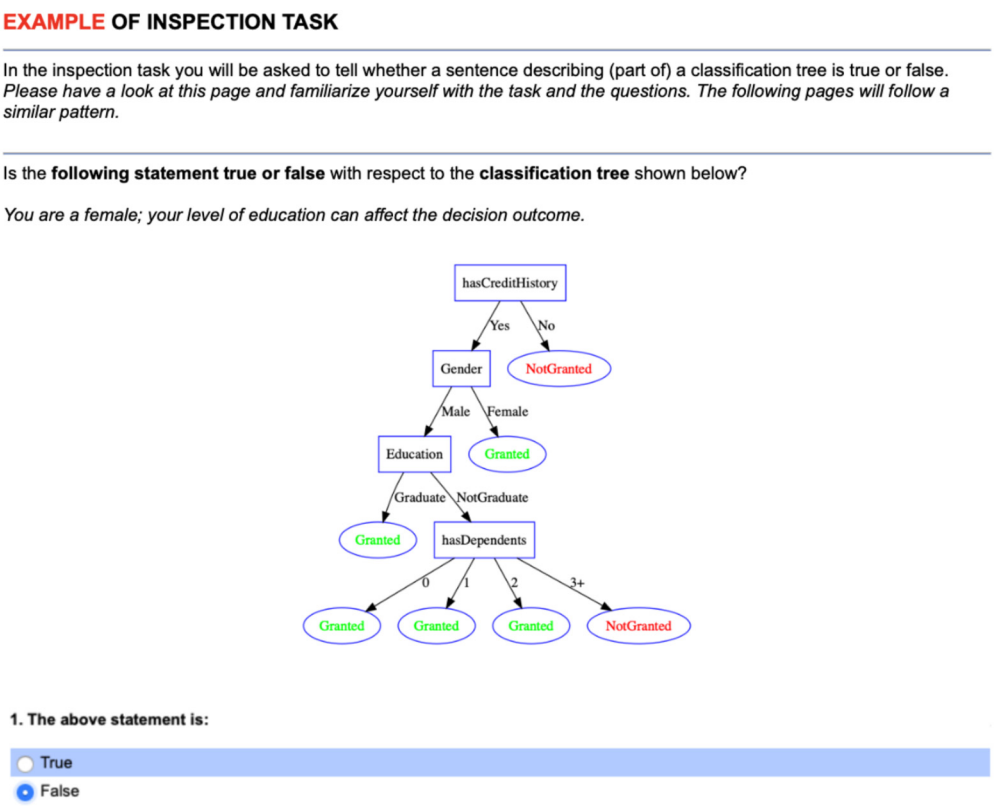

Fig. 4. Example of inspection task. Respondents were asked about the truth value of a given statement (e.g., "You are female; your level of education can affect the decision outcome.") given a decision tree. The respondent did not know whether the tree was generated using the ontology or not. For each respondent the reaction time and the accuracy of the answer were stored. After answering, respondents were asked to rate the confidence in their answer and the subjective understandability of the decision tree.

tively. In total, for each of the neural networks, we constructed six decision trees, varying their size (small, medium, and large as defined above), and whether or not an ontology had been used in extracting them. In this manner, we obtained a total of twelve decision trees (2 domains $\times$ 3 sizes $\times$ 2 ontology presence values). Fig. 2 shows two examples of extracted decision trees. The fidelity (proportion of agreement with the neural network) average over the extracted trees was 92.73% (TREPAN) 92.63% (TREPAN Reloaded) and 89.23% (TREPAN) 88.17% (TREPAN Reloaded) for the loan and heart dataset respectively (see also Table 3).

Procedure. The experiment used two online questionnaires on the usage of decision trees.[6] The questionnaires contained an introductory and an experimental phase.

In the introductory phase, subjects were shown a short video about decision trees, and how they are used for classification. In this phase, participants were asked to provide information on their age, gender, education, and on their familiarity with decision trees.

The experiment was subdivided into four tasks: classification, inspection, comparison and empowerment. Each task starts with an instruction page describing the task to be performed. In the first two tasks the participants were presented with the six trees corresponding to one of the two domains. In the classification task, subjects were asked to use a decision tree to assign one of two classes to a given case whose features are reported in a table (e.g., *Will the bank grant a loan to a male person, with 2 children, and a yearly income greater than €50.000,00?*). In the inspection task, participants had to decide on the truth value of a particular statement (e.g., *You are a male; your level of education affects your eligibility for a loan.*). The main difference between the two types of questions used in the two tasks is that the former provides all details necessary for performing the decision, whereas the latter only specifies whether a subset of the features influence the decision. In these two tasks, we recorded for each tree:

- Correctness of the response.

[6] In the following we will present some screenshots of the tasks the respondents were asked to carry out in the questionnaires. The complete questionnaires can be found as supplementary material.

Fig. 5. Example of comparison task. Respondents were asked to explicitly compare the understandability of two decision trees, one built with the use of the ontology and one built without its use. The respondents were not aware of the ontology use in the tree generation.

- Confidence in the response, as provided on a scale from 1 to 5 ('Totally not confident'=1, ..., 'Very confident'=5).
- Response time measured from the moment the tree was presented.
- Perceived tree understandability as provided on a scale from 1 to 5 ('Very difficult to understand'=1, ..., 'Very easily understandable'=5).

Figs. 3 and 4 show the user interface with an example of classification and inspection task respectively.

In the third task, the comparison, subjects were presented with a pair of trees of the same size, one extracted by TREPAN and one extracted by TREPAN Reloaded. Subjects were asked to rate on a five point scale which of the two trees is easier to understand. Each subject was presented with all three pairs of trees (small, medium, and large as defined above) in the selected domain. Fig. 5 shows an example of the comparison task.

In the fourth task, the empowerment task, respondents were shown a particular statement and had to decide what action (if any) they could take to change the decision of the model (e.g., *You are a male and you have 3 children. According to the decision tree, you are not eligible for a loan. Is there an action that you could take to become eligible?*). The empowerment task was designed to understand whether an explanation lets a subject identify what action could be taken to change the decision outcome. Respondents were asked to carry out this task with a tree extracted by TREPAN and a tree extracted by TREPAN Reloaded. Fig. 6 shows an example of the empowerment task.

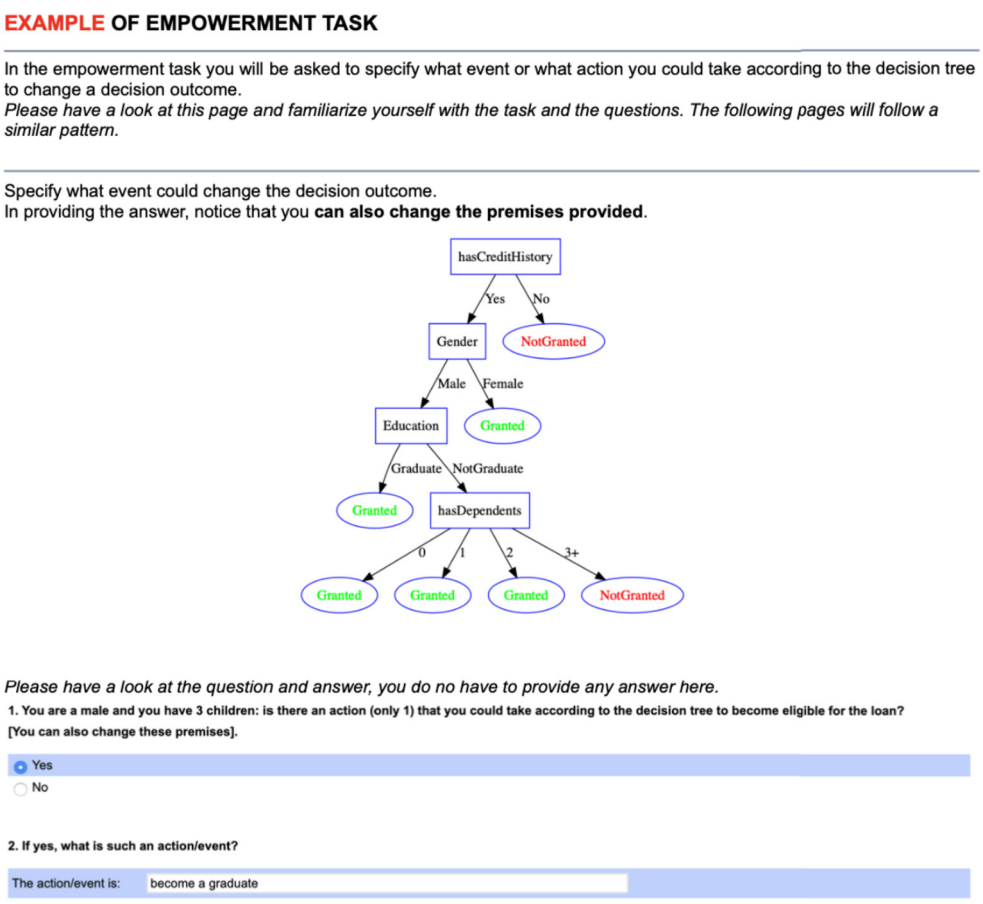

Fig. 6. Example of empowerment task. Given a certain context (e.g., "You are a male and you have 3 children") which can possibly lead to an outcome (e.g., "not eligible for a loan"), respondents were asked to say which action (if any) would lead to a different outcome (e.g., "Is there an action that you could take according to the decision tree to become eligible for the loan?"). The empowerment task was designed to understand whether an explanation lets a subject identify what action could be taken to change a given decision outcome.

Participants. 63 participants (46 females, 17 males) volunteered to take part in the experiment via an online survey.[7] Of these, 34 were exposed to trees from the finance domain and 29 to those in the medical domain. The average age of the participants was 33 ($\pm$ 12.23) years (range: 19–67). In terms of education, the highest level was a Ph.D. for 28 of them, a Master's degree for 9 of them, a Bachelor's for 12, and a high school diploma for 14. 47 of the respondents reported some familiarity with the notion of decision trees, while 16 reported no such familiarity.

5.2. Results

Classification and inspection tasks. We fitted a mixed-effects logistic regression model [3] predicting the correctness of the responses in the classification and inspection tasks. The independent fixed-effect predictors were the syntactic complexity of the tree, the presence or absence of an ontology in the tree generation (TREPAN vs. TREPAN-Reloaded), the task (classification vs. inspection), and the domain (financial vs. medical), as well as all possible interactions between them, as well as a random effect of the identity of the participant. A backwards elimination of factors revealed significant main effects of the task, indicating that responses were more accurate in the classification task than they were in the inspection ($z = -3.00$, $p = .0027$), of the syntactic complexity ($z = -3.47$, $p = .0005$), by which more complex tree produced less accurate responses, and of the presence of the ontology ($z = 3.70$, $p = .0002$), indicating that trees generated using the ontology indeed produced more accurate responses as shown in Fig. 7a. We did not observe any significant effects or interactions of the domain. The estimated effect of the syntactic complexity on accuracy is plotted in Fig. 7b. Interestingly, we did not observe any significant effects on the accuracy from the interaction between the syntactic complexity and the presence or absence of ontology in the tree generation.

[7] The participants were recruited among students as well as friends and acquaintances of the authors.

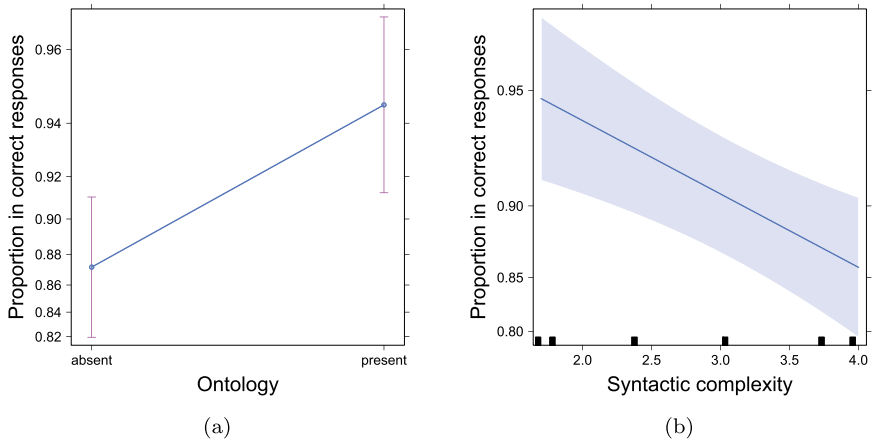

Fig. 7. Estimated main effects ontology presence (a) and of syntactic complexity (b) on the accuracy of subjects' responses.

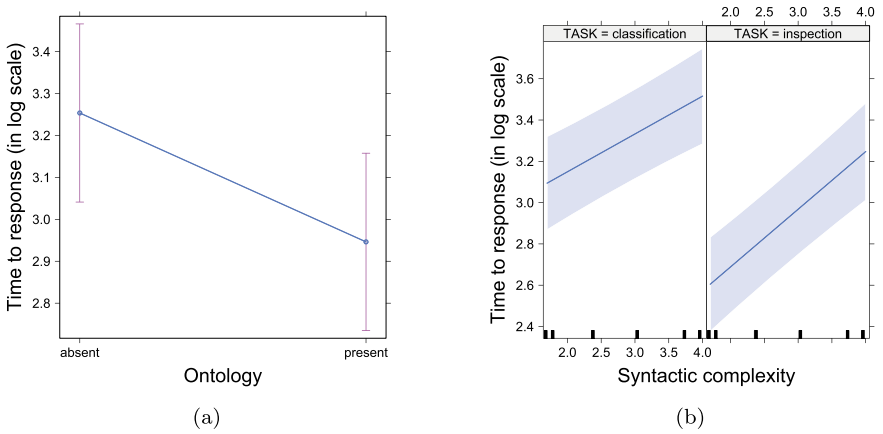

Fig. 8. Estimated main effects of ontology presence (a) and estimated two-way interactions between task and syntactic complexity on response time (b).

We analysed the reaction time of the correct responses in a linear mixed-effect regression model [3], with the log response time as the independent variable. As before, we included as possible fixed effects the task (classification vs inspection), the domain (medical vs financial), the syntactic complexity of the tree, and the presence or absence of ontology in the trees' generation, as well as all interactions between them. In addition, we included the identity of the participant as a random effect. A stepwise elimination of factors revealed main effects of task ($F(1, 593.87) = 20.81$, $p < .0001$), syntactic complexity ($F(1, 594.51) = 92.42$, $p < .0001$), ontology presence ($F(1, 594.95) = 51.75$, $p < .0001$), as well as significant interactions between task and syntactic complexity ($F(1, 594.24) = 4.06$, $p = .0044$), and task and domain ($F(2, 107.48) = 5.03$, $p = .0008$). Fig. 8b plots the estimated interaction between syntactic complexity and task on the response times. Overall, across both cases the more complex trees result in longer response times (as was evidenced by the main effect of syntactic complexity). The interaction indicates that this effect is significantly more marked in the inspection task than it is in the classification task. This is in line with our intuition that the inspection task requires a more detailed examination of the decision tree, and it is therefore more sensitive to its complexity. Crucially, we did not observe any significant effects of the interaction between syntactic complexity and the presence or absence of ontology in the trees' generation on the response time. In line with what we observed in the accuracy analysis, we find that those trees that were generated using an ontology were processed faster than those that were generated without one (see Fig. 8a).

We analysed the user confidence ratings using a linear mixed-effect regression model, with the confidence rating as the independent variable. We included as possible fixed effects the task (classification vs inspection), the domain (medical vs financial), the size of the tree, and the presence or absence of ontology in the trees' generation, as well as all interactions between them. In addition, we included the identity of the participant as a random effect. A step-wise elimination of factors revealed a main effect of ontology presence ($F(1, 689) = 14.38$, $p = .0002$), as well as significant interactions between task and syntactic complexity ($F(2, 689) = 46.39$, $p < .0001$), and task and domain ($F(2, 110.67) = 3.11$, $p = .0484$). These results are almost identical to what was observed in the response time analysis: users show more confidence on judgements performed on trees that involved an ontology, the effect of syntactic complexity is most marked in the inspection task, and the difference between domains only affects the classification task.

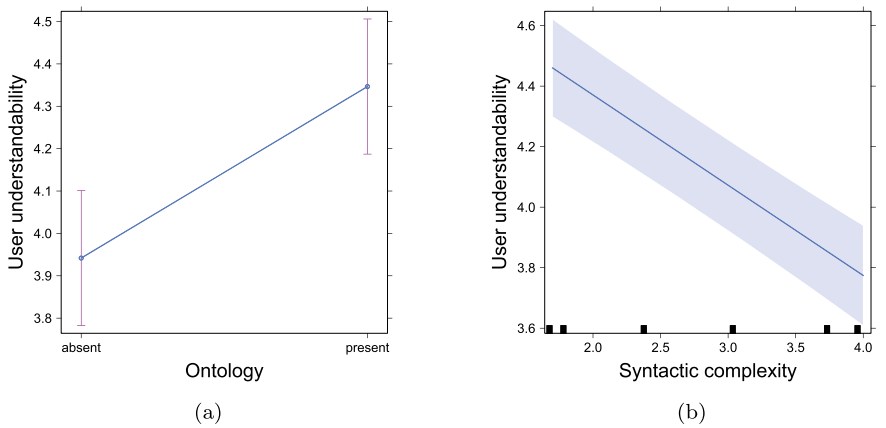

Fig. 9. Estimated main effects of ontology presence (a) and syntactic complexity (b) on understandability reported by users.

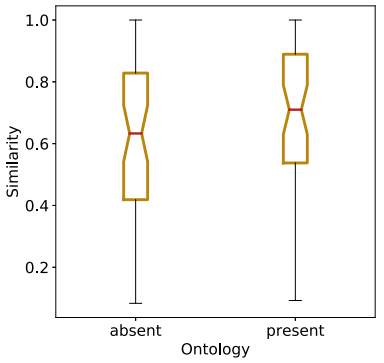

Fig. 10. Box-plot of ontology effect on semantic similarity of users' response w.r.t. correct answers in the Empowerment Task.

Finally, we also analysed the understandability ratings using a linear mixed-effect regression model, with the understandability rating as the independent variable. We included as possible fixed effects the task (classification vs inspection), the domain (medical vs financial), the syntactic complexity of the tree, and the presence or absence of ontology in the trees' generation, as well as all possible interactions between them. In addition, we included the identity of the participant as a random effect. A stepwise elimination of factors revealed significant main effects of task ($F(1, 690) = 27.21$, $p < .0001$), syntactic complexity ($F(1, 690) = 104.67$, $p < .0001$), and of the presence of an ontology ($F(1, 690) = 39.90$, $p < .0001$). Remarkably, we did not observe any significant effects of the interaction between the syntactic complexity and the presence or absence of ontology in the trees' generation on the users' reported understandability. These results are in all relevant aspects almost identical to what was observed in the accuracy analysis: the inspection task is harder, trees originating from an ontology are perceived as more understandable (see Fig. 9a), and more syntactically complex trees are less understandable than less complex ones (see Fig. 9b).

Comparison task. We analysed the comparison task to understand two things: first, we wanted to check what trees were considered more understandable in a 'task free' context; second, we wanted to check how many trees were rated in a similar way in the classification/inspection and comparison task. We found that the highest rated trees are those created by taking into account the ontology. A Wilcoxon signed rank test on matched pairs showed that the results are significantly different both w.r.t. the syntactic complexity of a tree ($p < .01$) and the use of an ontology ($p < .001$). Regarding the rating of trees in the comparison tasks w.r.t. the classification and inspection task, we found quite a good agreement in how the understandability of the tree is perceived as a response after having used it in a task, and as a response of a merely visual stimulus. In particular, in the loan domain, rates are aligned in 60.29%, 83.82%, 70.59% of the cases for small, medium and large trees respectively. In the heart domain, rates are aligned in 72.41%, 79.31%, 68.97% of the cases.

We fitted a mixed-effects logistic regression model to predict whether a user would deem trees extracted with Trepan Reloaded more understandable. The independent fixed-effect predictor was the tree size, as well as a random effect of the identity of the participant. The model revealed significant main effects of the tree size, indicating that decision trees extracted by Trepan Reloaded are deemed more understandable in the case of small ($z = 2.69$, $p = .0007$) and medium size trees ($z = 2.98$, $p = .0002$) respectively. In addition, when the domain is added as fixed effect, we observed a significant interaction of large size trees in the loan domain ($z = 2.05$, $p = .04$).

Table 1
Syntactic complexity (Eq. (2)) for trees inferred using C4.5, and extracted using TREPAN, and TREPAN Reloaded respectively.

	C4.5	TREPAN	TREPAN Reloaded
heart	5.64	3.56	**3.46**
loan	5.9	2.89	**2.63**

Table 2
Mean values of correct answers, time of response, user confidence, and user understandability for trees extracted using TREPAN and TREPAN Reloaded (standard deviations are reported in parentheses). The difference in results is statistically significant w.r.t. Mann-Whitney and Wilcoxon tests for all measures.

Task	Measure	TREPAN	TREPAN Reloaded
Class.	%Correct Responses	0.87 (0.33)	**0.94** (0.22)
	Time (sec)	43.25 (65.40)	**24.29** (15.86)
	Confidence	4.38 (0.87)	**4.56** (0.81)
	User Understandability	4.06 (1.01)	**4.50** (0.73)
Insp.	%Correct Responses	0.78 (0.42)	**0.90** (0.29)
	Time (sec)	28.35 (26.56)	**21.25** (39.58)
	Confidence	4.10 (0.99)	**4.36** (0.81)
	User Understandability	3.83 (1.06)	**4.20** (0.91)
Empow.	%Correct Responses	0.49 (0.50)	**0.67** (0.47)
	Time (sec)	117.46 (127.46)	**57.83** (37.96)

Empowerment task. To study the effects of the ontology in the empowerment task, we analysed the answers provided by the respondents, the accuracies of the answers, and the time of response.

We measured how similar the answers given by the respondents were to the correct responses. To this end, we made use of *spaCy*,[8] an industrial library for NLP tasks. SpaCy provides pre-trained word embeddings. We used these to convert the text of correct responses and of respondents' answers into vectors. With those vectors we calculated a semantic (cosine) similarity value. The analysis shows that users' responses are semantically closer to the correct answers when the decision trees were extracted by TREPAN Reloaded (Fig. 10).

We fitted a mixed-effects logistic regression model to predict the correctness of the responses. The independent fixed-effect predictors were the presence or absence of an ontology in the tree generation, the domain (financial vs. medical), all interactions between them, and a random effect of the identity of the participant. A backwards elimination of factors revealed significant effects of the domain identity indicating that trees generated using the ontology produced more accurate responses in the loan domain ($z = 2.50$, $p = 0.01$).

We analysed the response times of the correct responses using a linear mixed-effect regression model, with the log response time as the independent variable. We included as possible fixed effects the domain (medical vs. financial) and the presence or absence of ontology in the trees' generation, as well as the interactions between them. In addition, we included the identity of the participant as a random effect. A stepwise elimination of factors revealed main effects of the ontology presence ($F(1, 30.59) = 13.10$, $p = 0.001$), while we did not observe any significant interaction or effect of the domain. These results are identical to what observed in the classification and inspection task: trees extracted with an ontology lead to more accurate answers and shorter response times.

6. Discussion

We designed TREPAN Reloaded to use ontologies for the selection of features for conditions in split nodes, as described above, expecting it to lead to decision trees that are easy to understand. This ease of understanding was measured theoretically using a syntactic complexity measure, and empirically through time and accuracy of users' responses as well as subjective confidence and understandability.

First of all, the syntactic complexity (Eq. (2)) of the trees extracted with TREPAN Reloaded is slightly lower than of those generated with standard TREPAN (see Table 1). This small reduction of syntactic complexity may or may not lead to differences in the understandability of the extracted trees by people. However, in our experiments, both online implicit measures (accuracy and response time) and off-line explicit measures (user confidence and understandability ratings) show that trees generated with an ontology are significantly more accurately and quickly understood by subjects than trees generated without ontology in all measurements (see Table 2). Even when syntactic complexity is used as a separate independent variable in linear or logistic regression models as in the previous section (Section 5), there is always a significant positive influence of the ontology on understandability.

[8] https://spacy.io.

Table 3
Test-set accuracies and fidelities for trees extracted using TREPAN and TREPAN Reloaded.

	Accuracy				Fidelity	
	C4.5	NN	TREPAN	TREPAN Rld.	TREPAN	TREPAN Rld.
heart	81.97%	94.65%	86.02%	82.80%	89.25%	88.17%
loan	80.48%	85.98%	82.43%	80.87%	92.75%	92.64%

As we anticipated, forcing the outputs of TREPAN onto a pre-determined ontology as in TREPAN reloaded impacts the fidelity (and accuracy) of the resulting trees (see Table 3). However, the small compromise in the fidelity of the neural network reconstruction (in both examples a drop of around one percent) will in many applications be more than compensated for by the substantial improvement in the ease with which humans can understand the resulting trees.

At this point, one might wonder why we should bother to create surrogate decision trees from black-box models, rather than inferring them directly from data. As already noticed in the original TREPAN work [17], extracted trees from networks can actually result in better trees than those one would obtain by building the decision trees directly. To demonstrate this point, we also trained decision trees directly from the datasets using the classical C4.5 algorithm. Table 3 shows that the trees inferred by the TREPAN variants are as accurate – if not more – than those inferred directly. Moreover, the trees built directly had syntactic complexities that roughly doubled those of the trees extracted using either TREPAN variant (see Table 1). This indicates that constructing trees directly from the data results in trees substantially more complex than those extracted by TREPAN variants, that nevertheless do not outperform them in the task.

There is a similarly small compromise in the accuracy of the decision trees (see Table 3). As we discussed above, in this approach the accuracy of the resulting trees (i.e., their ability to match ground truth) is less relevant than their fidelity (i.e., their ability to match the behaviour of the model we intend to explain). Nevertheless, our TREPAN Reloaded method improves the understandability of the trees w.r.t. the original TREPAN, while compromising little on the accuracy.

Apart from improving the understandability of (extracted) decision trees, ontologies also pave the way towards the capability of changing the level of abstraction of explanations to match different user profiles or audiences. For instance, the level of technicality used in an explanation for a medical doctor or a fund manager should not be the same as that used for patients or banking customers. Therefore it is desirable to adapt explanations without changing the underlying explanation procedure. Ontologies are amenable to automated abstractions to improve understandability [34]. The idea of concept refinement adopted here can be extended to operate on changing the definition of concepts and make them more general or more specific by means of refinement operators [10,64,52]. This is a line of work that we find a natural continuation of the current study.

In its current form, TREPAN Reloaded requires a predefined ontology onto which the features used by our algorithm should be mapped. In such cases, which are common in many domains (e.g., medical, pharmaceutical, legal, biological, etc.), one can directly apply TREPAN Reloaded to improve the quality of the explanations. Additional work beyond the scope of the current study, would be to automatically construct the most appropriate ontology to be mapped onto. Such a process could be achieved by automatically mapping sets of features into pre-existing general domain ontologies (e.g., MS Concept Graph [69], DBpedia [38], Freebase [6], YAGO [61], WordNet [42]). Many approaches have been proposed for extracting the schema of the tables, and mapping it to existing ontologies. For instance, Mulwad et al. [44,45] have made significant contribution for interpreting tabular data using linked open data coming from independent domains. They proposed several approaches that use background knowledge from pre-existing general domain ontologies to infer the semantics of column headers, table cell values and relations between columns and represent the inferred meaning as graph of RDF triples.

The provision of some form of explicit knowledge, rather than being particular to our method, resides at the core of any attempts at human interpretable explanations. Whether such knowledge is in the form of a domain-specific ontology (as in this study), or as a domain-general one to be adapted ad-hoc, will depend on the particulars of specific applications. This limits the application in cases where features do not have a meaning specified in an ontology, as it is necessary to build the semantic layer needed by our approach. However, this can even be done in cases where there are no prior concepts, as long as the concept can be structured and communicated to the user. For instance, in the case of image classification using CNNs, where pixels do not have any semantics associated per se, features learned by filters in convolutional layer could be associated to semantic concepts. These concepts could then be used to create an interpretable model explaining what the CNNs learned.

The method proposed in this paper and other approaches (e.g., LIME [55], SHAP [60,40]) explain the role of features in decision making by creating surrogate models that are interpretable. These models usually have a low number of features in order to facilitate interpretation of the model. This approach becomes less effective for higher-dimensional feature spaces as there is an increasing trade-off between fidelity and interpretability. Here one could take advantage of domain knowledge and ontologies to group features into more abstract concepts, and use those to present more concise explanations.

7. Related work

Most approaches to interpretable machine learning focus either on building directly interpretable models, or on reconstructing *post-hoc* local or global explanations [26]. Whilst local explanation methods aim to explain specific decisions,

global approaches aim to provide a more comprehensive understanding of how a model works. Our approach belongs to the category of post-hoc global explanation methods.

In local explanation methods, the individual predictions of a black-box model is approximated by generating local surrogate models that are intrinsically interpretable. This strategy has been implemented for instance in the popular Local Interpretable Model-agnostic Explanations (LIME) [55]. LIME perturbs the data for a given decisions and, after feeding the black-box model with perturbed data, creates a new linear model from the predictions. The linear model is reduced to k factors using Lasso, generating a small set of coefficients that constitute the explanation. The creation of an explainable surrogate is similar to our approach, but while LIME builds local surrogate models, our explanation is global and non-linear. A more recent approach to linear surrogate models is CLEAR: Counterfactual Local Explanations via Regression [66]. CLEAR provides counterfactual explanations by building on and extending two boundary (b-)counterfactuals and LIME. CLEAR provides b-counterfactuals that are explained by regression coefficients including interaction terms, evaluates the fidelity of its local explanations to the underlying learning system, and uses the values of b-counterfactual. This achieves a better fidelity with respect to the black-box models' decision boundaries. Several local explanation methods have been proposed based on Shapley values (e.g., [40,60]), which quantify the influence of a feature on a decision by training models on all subsets of features and observing their output depending on the presence of the given feature. Our method does not immediately provide feature importance values for individual decisions, but these could be obtained by analysing the tree and decision boundaries are explicit in generated trees.

The authors in [47] present *Doctor XAI*, a local model-agnostic technique suitable to explain black-box classifiers that use multi-labelled, sequential, and ontology-linked data. In particular, they focus on explaining *Doctor AI*, a neural network classifier that predicts a patient's next visit time given the patient's clinical history. The authors propose an explanation pipeline to generate post-hoc local explanations of these predictions. Each explanation is presented as a decision rule, which is extracted from a decision tree that is trained using a synthetic set of neighbours of the instance, for which the decision should be explained. The set of neighbours is artificially created either by sampling them from a normal distribution, or by exploiting the ontology linked to the data (e.g., the ICD-9 ontology). In this latter case, the ontology is used to calculate a first set of real neighbours using the similarity between patients, visits, and code diseases stored in the ontology. Then, this set of real neighbours is perturbed, by masking all the occurrences of the items with the same least common superconcept, to generate synthetic neighbours. The synthetic neighbours are then used to train the decision tree. The authors show that explanations generated using the ontology can have a higher fidelity than those generated without the use of the ontology. Explanations' interpretability is measured by taking into account the syntactic complexity of the decision rules, and, thus, of the extracted trees. However, their human understandability is not directly tested.

In [53], concepts are used to group features and embed them into surrogate models in order to constrain the training. Concepts are created either by experts or by correlating existing features. The authors propose *ConceptTrees*, a version of Trepan that builds on features belonging to concepts in the extraction of a decision tree. They compare surrogate trees extracted by ConceptTree with those extracted by the original version of Trepan. To evaluate their approach they use FRED-MD, a macroeconomic database designed for empirical analysis of the US economy. The dataset comes with an associated taxonomy of the features that support the definition of concepts. While their results show that surrogate trees preserve accuracy and fidelity compared to original versions, the improvement in human understandability of explanations was not explicitly tested with users. As the authors point out, further research could involve a deeper assessment of their propositions, both quantitatively and qualitatively.

Dhurandhar et al. [20] propose *ProfWeight*, a method designed to transfer information from a pre-trained deep learning model to a simpler one, e.g., a simpler neural network or a decision tree. ProfWeight is part of the IBM 360 Explainability Toolkit [1]. The idea behind this approach is to attach and train (linear) probes on the internal layers (corresponding to intermediate representations) of a high performing neural network. The trained probes allow building confidence profiles over the classified examples, in such a way that harder examples are associated with weaker confidence profiles having lower confidence weights. These weights are used to create a weighted version of the original dataset that is used to retrain the simpler model. The weights are essentially informing the simpler model on which examples to focus to improve its generalisation performance. The authors show the benefits of their approach in two scenarios, namely, digit recognition using the CIFAR dataset and a semi-conductor manufacturing setting in which ProfWeight is used to improve the performances of decision trees. The main motivation behind this approach is not only to create more explainable decision models. It also aims at supporting domain experts who want to use a more familiar method (which usually has lower accuracy performances), to deal with the lack of lot of training data, or with limited computational resources, such as in the case of edge computing. Different to our approach, they assume to have a white-box access to the layers of a complex network model, whereas we treat the black-box as an oracle.

The authors in [4] propose a model extraction technique for interpreting the reasoning process performed by black-box models, in particular, random forests, and neural networks. The technique generates an interpretable approximation of the original model, in the form of a decision tree. To avoid overfitting, they use active learning to sample a large number of training points, and compute the corresponding labels using the complex model. Our approach is similar since Trepan also uses new sampled data during the extraction of the decision tree. The main motivation behind this approach is to extract global explanations to debug issues with random forests and neural networks – e.g., assessing the dependence on prejudiced features and determining why certain models perform worse –, or to understand the high-level structure of control policy learned using reinforcement learning. In the case of classification, they compare the performance of extracted trees with

CART and other knowledge extraction techniques and formats, among others rule lists, in classification, regression, and reinforcement learning. To evaluate the understandability of distilled decision trees they designed and ran a user study. The user study used subjects with machine learning and programming background. The subjects were asked to answer questions about classifying instances and identifying sub-populations for which a certain feature is relevant using decision trees and rule lists. In this respect the user study is similar to ours concerning Task 1 (classification) and Task 2 (inspection). The results show that users responded equally or more accurately using decision tree, even though trees were much larger than the rule lists. Besides, users experienced some difficulty with the conditional structure of the rule lists. In our study, we focused on the evaluation of the understandability of decision trees, but also on comparing two 'representation formats', where the format is understood as extracted trees with and without the use of an ontology. Similarly, [35,70] explain black-box models by means of approximate interpretable models, namely, decision sets and rule lists respectively. Lakkaraju et al. [35] propose a model agnostic framework for explaining the behaviour of black-box classifiers by simultaneously optimising for fidelity to the original model and interpretability of the explanation. The explanations are served as two-level decision sets. A user study shows that their model outperforms other explanation models in percentage of correct responses and time of response.

With a different scope, some works focus on building terminological decision trees from and with ontologies, e.g., [58,71]. Rizzo et al. [58] describe a framework to learn *Terminological Decision Trees* from examples encoded in an ontological knowledge base. Terminological decision trees are an extension of first order logic decision trees dealing with description logics. The internal nodes in a terminological decision tree contain DL concept descriptions. Each concept is used as a test to route individuals that are classified towards the leaves. Given the fact that they rely on explicit knowledge, terminological decision trees provide an interpretable classification model with a trade-off between predictiveness and comprehensibility. Each target concept is defined indeed as the disjunction (given the open-world assumption) of the conjunctions of the concept descriptions on the paths reaching the leaves labelled with the target concept. The use of these trees as a classification model is the prediction of the membership of unseen individuals (i.e., further individuals that were not considered in the learning phase) with respect to the target concept. As an extension of this framework, the authors present *Terminological Random Forests* [58]. Terminological Random Forests are based on ensemble-models built upon terminological decision trees to cope with overfitting and class imbalance. Zhang et al. [71] present an ontology-driven decision tree learning algorithm to learn classification rules at multiple levels of abstraction. The authors present some preliminary results to demonstrate the feasibility of the proposed approach, a general strategy for transforming traditional inductive learning algorithms into ontology-guided inductive learning algorithms for data-driven discovery of relationships at multiple levels of abstraction. These approaches are more focused on performing a classification task while building a tree rather than distilling a decision tree from a classification process computed by a black-box.

Other works showed how using open linked data is useful to explain data pattern discovered by data mining techniques. Ristoski and Paulheim [57] provide an overview of several works that exploit knowledge graphs to create human explanations for data mining and knowledge discovery processes. For instance, *Explain-a-LOD* [49] automatically generates hypothesis for explaining statistics by using ontologies. The tool uses *FeGeLOD* [50] to enhance statistical datasets with background information from DBpedia [38], and uses correlation analysis and rule learning for producing hypotheses which are presented to the user. Ilaria et al. [33] introduce *Dedalo*, a framework that dynamically traverses a knowledge graph to find commonalities that form explanations for items of a cluster. The authors were able to extract interesting and representative explanation for the clusters. However, the number of resulting atomic rules was rather large, and such rules have to be aggregated in a post-processing step. To cope with this, the same authors extended Dedalo using neural networks to predict whether two rules, if combined, can lead to the creation of a new improved rule. Lavrac et al. [36] propose to use background knowledge from semantically annotated biological data repositories to perform semantic subgroup discovery. In particular, they implemented *SEGS* a system that automatically formulates biological hypotheses. Namely, rules which define groups of differentially expressed genes.

Finally, the idea of concept refinement proposed here was formerly used in applications of computational creativity based on conceptual blending [24,9] and ontology repair [64,52]. In conceptual blending, new concepts are created by combining existing ones; first, the commonalities of the original concepts are searched; then, specifics of the original concept are projected into a new concept, a blend. Refinement operators were used to refine (generalise) the input concepts, until a common concept description is found [24,9]. The idea of concept refinement was extended to cope with axiom refinement, leading to the definition of axiom weakening [64,52,12]. In [64,52] the authors showed how axiom weakening is an effective way to repair inconsistent ontologies in a single and a multi-agent setting.

8. Conclusion and future works

In this paper, we proposed Trepan Reloaded, to improve human understandability of decision trees. It is an extension of Trepan, an algorithm that extracts global post-hoc explanations of black-box models in the form of decision trees. Our algorithm takes into account ontologies in the extraction of decision trees from black-box classifiers. Our approach is model-agnostic and can be applied to explain any black-box classifiers.

We showed how the use of ontologies improves the understanding of the extracted trees by actual users. We measured the understanding through a rigorous experimental evaluation through time and accuracy of human responses as well as reported user confidence and understandability. All our measures indicate that trees extracted by Trepan Reloaded are

significantly more accurately and more easily understood by subjects than trees generated by TREPAN, with only a small reduction of accuracy and the fidelity (see Section 6). The dataset and the analysis performed are published for reproducibility in [13].

The results we obtained are very promising and open up several directions for future research. Firstly, we plan to extend this work to support the generation of explanations that can accommodate different user profiles. Secondly, we aim to investigate applying our approach to domains where features might not have a clear semantics meaning such as in image classification (e.g., [72]). Lastly, we also believe that this approach can be useful in identifying and justifying algorithmic bias [27], in particular, to understand if any (undesirable) discrimination features are used in a black-box classifier. For instance, consider the 'gender' feature in the decision trees shown in Fig. 2. This feature could be considered a undesirable discriminatory feature depending on the decision domain. Explanations of black-boxes, in the form of extracted decision trees as in this paper, provide a means to identify biases in black-box models.

Extensions of previous work

This work is based on and extends the conference paper [14]. Compared to the conference paper, this paper contains the following differences:

- An extended 'preliminaries' section, providing more details on decision trees, and concept refinement, two core notions for our approach.
- A more comprehensive user study about the positive influence of ontologies on the understandability of explanations by human users. In particular, we present an extended analysis of Task 1 (Classification) and Task 2 (Inspection). We also present the analysis of two further tasks of the user study, which were not included in the conference paper: Task 3 (comparison), and Task 4 (empowerment).
- An extended and more detailed related work section situating our work w.r.t. existing approaches in the literature. The related work section emphasises that not many works exist that use ontologies to enhance the understandability of explanations or that study and measure human understandability of explanations.
- We published the data collected in the user study (anonymised), as well as the scripts created to analyse the data in Mendeley Data, so that our analysis is reproducible [13].
- We attached the questionnaires used in the user study as supplementary material.

Declaration of competing interest

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

Acknowledgements

A significant part of the work has been carried out at Alpha Health, Telefónica Innovación Alpha, Barcelona, Spain. The authors thank the reviewers for their valuable comments. The authors thank the Department of Innovation, Research and University of the Autonomous Province of Bozen/Bolzano for covering the Open Access publication costs.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.artint.2021.103471.

References

[1] V. Arya, R.K.E. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S.C. Hoffman, S. Houde, Q.V. Liao, R. Luss, a. MojsiloviĆ, S. Mourad, P. Pedemonte, R. Raghavendra, J. Richards, P. Sattigeri, K. Shanmugam, M. Singh, K.R. Varshney, D. Wei, Y. Zhang, One Explanation Does Not Fit All: a Toolkit and Taxonomy of AI Explainability Techniques, 2019.

[2] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, P.F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, New York, NY, USA, ISBN 0-521-78176-0, 2003.

[3] R. Baayen, D. Davidson, D. Bates, Mixed-effects modeling with crossed random effects for subjects and items, J. Mem. Lang. (ISSN 0749-596X) 59 (4) (2008) 390–412.

[4] O. Bastani, C. Kim, H. Bastani, Interpreting blackbox models via model extraction, CoRR, arXiv:1705.08504 [abs], 2017, http://arxiv.org/abs/1705.08504.

[5] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J.M.F. Moura, P. Eckersley, Explainable machine learning in deployment, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT*'20, New York, NY, USA, Association for Computing Machinery, ISBN 9781450369367, 2020, pp. 648–657.

[6] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD'08, New York, NY, USA, ISBN 9781605581026, 2008, pp. 1247–1250, Association for Computing Machinery.

[7] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth, ISBN 0-534-98053-8, 1984.

[8] B.G. Buchanan, E.H. Shortliffe, Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project, Addison-Wesley Longman Publishing Co., Inc., ISBN 0201101726, 1984.

[9] R. Confalonieri, O. Kutz, Blending under deconstruction: the roles of logic, ontology, and cognition in computational concept invention, Ann. Math. Artif. Intell. (ISSN 1012-2443) 88 (5) (2020) 479–516, https://doi.org/10.1007/s10472-019-09654-6.

[10] R. Confalonieri, M. Eppe, M. Schorlemmer, O. Kutz, R. Peñaloza, E. Plaza, Upward refinement operators for conceptual blending in the description logic $\mathcal{EL}^{++}$, Ann. Math. Artif. Intell. (ISSN 1012-2443) 82 (1–3) (2018) 69–99, https://doi.org/10.1007/s10472-016-9524-8, https://www.sciencedirect.com/science/article/abs/pii/S0952197615002006.

[11] R. Confalonieri, T.R. Besold, T. Weyde, K. Creel, T. Lombrozo, S. Mueller, P. Shafto, What makes a good explanation? Cognitive dimensions of explaining intelligent machines, in: Proc. of the 41st Annual Meeting of the Cognitive Science Society, CogSci 2019, 2019, https://mindmodeling.org/cogsci2019/papers/0013/index.html.

[12] R. Confalonieri, P. Galliani, O. Kutz, D. Porello, G. Righetti, N. Troquard, Towards even more irresistible axiom weakening, in: Proc. of the 33rd International Workshop on Description Logics, Online, DL 2020, September 12–14, 2020, in: CEUR Workshop Proceedings, vol. 2663, CEUR-WS.org, 2020, http://ceur-ws.org/Vol-2663/paper-8.pdf.

[13] R. Confalonieri, T. Weyde, T.R. Besold, F.M. del Prado Martín, Understandability of Global Post-hoc Explanations of Black-box Models: Dataset and Analysis, 2020, Mendeley Data.

[14] R. Confalonieri, T. Weyde, T.R. Besold, F.M. del Prado Martín, Trepan reloaded: a knowledge-driven approach to explaining black-box models, in: Proceedings of the 24th European Conference on Artificial Intelligence, in: Frontiers in Artificial Intelligence and Applications, vol. 325, IOS Press, 2020, pp. 2457–2464.

[15] R. Confalonieri, L. Coba, B. Wagner, T.R. Besold, A historical perspective of explainable artificial intelligence, WIREs Rev. Data Min. Knowl. Discov. 11 (1) (2021), https://doi.org/10.1002/widm.1391, https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1391.

[16] M.W. Craven, Extracting Comprehensible Models from Trained Neural Networks, PhD thesis, the University of Wisconsin-Madison, 1996, AAI9700774.

[17] M.W. Craven, J.W. Shavlik, Extracting tree-structured representations of trained networks, in: NIPS 1995, MIT Press, 1995, pp. 24–30.

[18] DARPA, Explainable AI – program, https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf, 2016.

[19] A.S. d'Avila Garcez, K. Broda, D.M. Gabbay, Symbolic knowledge extraction from trained neural networks: a sound approach, Artif. Intell. (ISSN 0004-3702) 125 (1–2) (2001) 155–207.

[20] A. Dhurandhar, K. Shanmugam, R. Luss, P.A. Olsen, Improving simple models with confidence profiles, in: NIPS 2018, Curran Associates, Inc., 2018, pp. 10296–10306, http://papers.nips.cc/paper/8231-improving-simple-models-with-confidence-profiles.pdf.

[21] F.C. Donders, On the speed of mental processes, Acta Psychol. 30 (1969) 412–431.

[22] F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning, CoRR (2017), arXiv:1702.08608 [abs].

[23] Eleanor, E. Rosch, B. Carolyn, C.B. Mervis, W.D. Gray, D.M. Johnson, J.L. Penny, P. Boyes-Braem, Basic objects in natural categories, Cogn. Psychol. 8 (1976) 382–439.

[24] M. Eppe, E. Maclean, R. Confalonieri, O. Kutz, M. Schorlemmer, E. Plaza, K.-U. Kühnberger, A computational framework for conceptual blending, Artif. Intell. (ISSN 0004-3702) 258 (105–129) (2018), https://doi.org/10.1016/j.artint.2017.11.005, https://www.sciencedirect.com/science/article/pii/S000437021730142X.

[25] A.D. Garcez, T.R. Besold, L. De Raedt, P. Foldiak, P. Hitzler, T. Icard, K.U. Kiihnberger, L.C. Lamb, R. Miikkulainen, D.L. Silver, Neural-symbolic learning and reasoning: contributions and challenges, in: AAAI Spring Symposium – Technical Report, ISBN 9781577357070, 2015.

[26] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. (ISSN 0360-0300) 51 (5) (2018) 1–42.

[27] S. Hajian, F. Bonchi, C. Castillo, Algorithmic bias: from discrimination discovery to fairness-aware data mining, in: B. Krishnapuram, M. Shah, A.J. Smola, C.C. Aggarwal, D. Shen, R. Rastogi (Eds.), Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 13-17, 2016, ACM, San Francisco, CA, USA, 2016, pp. 2125–2126.

[28] M. Hind, Explaining explainable AI, XRDS (ISSN 1528-4972) 25 (3) (2019) 16–19, https://doi.org/10.1145/3313096, http://doi.acm.org/10.1145/3313096.

[29] R.R. Hoffman, S.T. Mueller, G. Klein, J. Litman, Metrics for explainable AI: challenges and prospects, CoRR (2018), arXiv:1812.04608 [abs].

[30] A. Holzinger, Interactive machine learning for health informatics: when do we need the human-in-the-loop?, Brain Inform. (ISSN 2198-4018) 3 (2016) 119–131, https://doi.org/10.1007/s40708-016-0042-6, http://www.springer.com/computer/ai/journal/40708.

[31] A. Holzinger, M. Plass, M. Kickmeier-Rust, K. Holzinger, G.C. Crişan, C.-M. Pintea, V. Palade, Interactive machine learning: experimental evidence for the human in the algorithmic loop, Appl. Intell. (ISSN 0924-669X) 49 (7) (2019) 2401–2414, https://doi.org/10.1007/s10489-018-1361-5.

[32] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, B. Baesens, An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models, Decis. Support Syst. (ISSN 0167-9236) 51 (1) (2011) 141–154.

[33] I. Tiddi, M. d'Aquin, E. Motta, Dedalo: looking for clusters explanations in a labyrinth of linked data, in: V. Presutti, C. d'Amato, F. Gandon, M. d'Aquin, S. Staab, A. Tordai (Eds.), The Semantic Web: Trends and Challenges, Springer International Publishing, Cham, ISBN 978-3-319-07443-6, 2014, pp. 333–348.

[34] C.M. Keet, Enhancing comprehension of ontologies and conceptual models through abstractions, in: Proc. of the 10th Congress of the Italian Association for Art. Intel., AI*IA 2007, 2007, pp. 813–821.

[35] H. Lakkaraju, E. Kamar, R. Caruana, J. Leskovec, Interpretable & explorable approximations of black box models, CoRR (2017), arXiv:1707.01154 [abs], http://arxiv.org/abs/1707.01154.

[36] N. Lavrac, A. Vavpetic, L.N. Soldatova, I. Trajkovski, P.K. Novak, Using ontologies in semantic data mining with SEGS and g-SEGS, in: T. Elomaa, J. Hollmén, H. Mannila (Eds.), Proceedings of the 14th Int. Conf of Discovery Science, DS 2011, in: LNCS, vol. 6926, Springer, ISBN 978-3-642-24476-6, 2011, pp. 165–178.

[37] J. Lehmann, P. Hitzler, Concept learning in description logics using refinement operators, Mach. Learn. (ISSN 0885-6125) 78 (1–2) (2010) 203–250.

[38] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P.N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, C. Bizer, DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia, J. Web Semant. 6 (2) (2015) 167–195, http://jens-lehmann.org/files/2015/swj_dbpedia.pdf.

[39] Z.C. Lipton, The mythos of model interpretability, Queue (ISSN 1542-7730) 16 (3) (June 2018) 30:31–30:57.

[40] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 4765–4774.

[41] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, CoRR, arXiv:1908.09635 [abs], 2019, http://arxiv.org/abs/1908.09635.

[42] G.A. Miller, Wordnet: a lexical database for English, Commun. ACM 38 (11) (Nov. 1995) 39–41, https://doi.org/10.1145/219717.219748.

[43] T. Miller, Explanation in artificial intelligence: insights from the social sciences, Artif. Intell. (ISSN 0004-3702) 267 (2019) 1–38, https://doi.org/10.1016/j.artint.2018.07.007, http://www.sciencedirect.com/science/article/pii/S0004370218305988.

[44] V. Mulwad, T. Finin, A. Joshi, Semantic message passing for generating linked data from tables, in: Proceedings of the 12th International Semantic Web Conference – Part I, ISWC'13, Springer-Verlag, Berlin, Heidelberg, ISBN 9783642413346, 2013, pp. 363–378, https://doi.org/10.1007/978-3-642-41335-3_23.

[45] V. Mulwad, T. Finin, A. Joshi, Interpreting medical tables as linked data for generating meta-analysis reports, in: Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration, IEEE IRI 2014, Aug. 2014, pp. 677–686.

[46] I. Nunes, D. Jannach, A systematic review and taxonomy of explanations in decision support and recommender systems, User Model. User-Adapt. Interact. (ISSN 1573-1391) 27 (3-5) (2017) 393–444, https://doi.org/10.1007/s11257-017-9195-0, http://link.springer.com/10.1007/s11257-017-9195-0http://ls13-www.cs.tu-dortmund.de/homepage/publications/jannach/Journal_UMUAI_2017_2.pdf.
[47] C. Panigutti, A. Perotti, D. Pedreschi, Doctor XAI: an ontology-based approach to black-box sequential data classification explanations, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, FAT**'20, New York, NY, USA, ISBN 9781450369367, 2020, pp. 629–639, Association for Computing Machinery.
[48] Parliament and Council of the European Union. General Data Protection Regulation, 2016.
[49] H. Paulheim, Explain-a-LOD: using linked open data for interpreting statistics, in: Proc. of the 2012 ACM Int. Conf. on Intelligent User Interfaces, IUI'12, ACM, ISBN 9781450310482, 2012, pp. 313–314.
[50] H. Paulheim, J. Fümkranz, Unsupervised generation of data mining features from linked open data, in: Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics, WIMS'12, New York, NY, USA, ISBN 9781450309158, 2012, Association for Computing Machinery.
[51] R. Piltaver, M. Luštrek, M. Gams, S. Martinčić-Ipšić, What makes classification trees comprehensible?, Expert Syst. Appl. (ISSN 0957-4174) 62 (C) (Nov. 2016) 333–346.
[52] D. Porello, N. Troquard, R. Peñaloza, R. Confalonieri, P. Galliani, O. Kutz, Two approaches to ontology aggregation based on axiom weakening, in: J. Lang (Ed.), Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, ISBN 978-0-9992411-2-7, 2018, pp. 1942–1948, ijcai.org.
[53] X. Renard, N. Woloszko, J. Aigrain, M. Detyniecki, Concept tree: high-level representation of variables for more interpretable surrogate decision trees, CoRR, arXiv:1906.01297 [abs], 2019, http://arxiv.org/abs/1906.01297.
[54] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in: IJCAI 1995, Morgan Kaufmann Publishers Inc., ISBN 1-55860-363-8, 1995, pp. 448–453, 978-1-558-60363-9.
[55] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": explaining the predictions of any classifier, in: Proc. of the 22nd Int. Conf. on Knowledge Discovery and Data Mining, KDD'16, ACM, ISBN 978-1-4503-4232-2, 2016, pp. 1135–1144.
[56] M.T. Ribeiro, S. Singh, C. Guestrin, Anchors: high-precision model-agnostic explanations, in: AAAI, AAAI Press, 2018, pp. 1527–1535.
[57] P. Ristoski, H. Paulheim, Semantic Web in data mining and knowledge discovery: a comprehensive survey, J. Web Semant. (ISSN 1570-8268) 36 (2016) 1–22, https://doi.org/10.1016/J.WEBSEM.2016.01.001, https://www.sciencedirect.com/science/article/pii/S1570826816000020.
[58] G. Rizzo, C. d'Amato, N. Fanizzi, F. Esposito, Tree-based models for inductive classification on the web of data, J. Web Semant. (ISSN 1570-8268) 45 (2017) 1–22.
[59] D. Sánchez, M. Batet, D. Isern, Ontology-based information content computation, Knowl.-Based Syst. (ISSN 0950-7051) 24 (2) (2011) 297–303.
[60] L.S. Shapley, Notes on the n-Person Game—I: Characteristic-Point Solutions of the Four-Person Game, RAND Corporation, Santa Monica, CA, 1951.
[61] F.M. Suchanek, G. Kasneci, G. Weikum, Yago: a core of semantic knowledge, in: Proceedings of the 16th International Conference on World Wide Web, WWW'07, ACM, ISBN 978-1-59593-654-7, 2007, pp. 697–706.
[62] N. Tintarev, J. Masthof, Explaining recommendations: design and evaluation, in: Recommender Systems Handbook, Springer US, Boston, MA, ISBN 978-1-4899-7637-6, 2015, pp. 217–253, http://link.springer.com/10.1007/978-1-4899-7637-6_10, http://www.springerlink.com/index/10.1007/978-0-387-85820-3.
[63] G.G. Towell, J.W. Shavlik, Extracting refined rules from knowledge-based neural networks, Mach. Learn. (ISSN 0885-6125) 13 (1) (1993) 71–101.
[64] N. Troquard, R. Confalonieri, P. Galliani, R. Peñaloza, D. Porello, O. Kutz, Repairing ontologies via axiom weakening, in: S.A. McIlraith, K.Q. Weinberger (Eds.), Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18, New Orleans, Louisiana, USA, February 2–7, 2018, AAAI Press, New Orleans, Louisiana, USA, 2018, pp. 1981–1988, https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17189.
[65] P.R. van der Laag, S.-H. Nienhuys-Cheng, Completeness and properness of refinement operators in inductive logic programming, J. Log. Program. (ISSN 0743-1066) 34 (3) (1998) 201–225.
[66] A. White, A.S. d'Avila Garcez, Measurable counterfactual local explanations for any classifier, in: G.D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín, J. Lang (Eds.), ECAI 2020 – 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, in: Frontiers in Artificial Intelligence and Applications, vol. 325, IOS Press, 2020, pp. 2529–2535.
[67] M.R. Wick, W.B. Thompson, Reconstructive expert system explanation, Artif. Intell. (ISSN 0004-3702) 54 (1–2) (Mar. 1992) 33–70.
[68] J.B. William Lidwell, Kritina Holden, Universal Principles of Design, Rockport, 2003.
[69] W. Wu, H. Li, H. Wang, K.Q. Zhu, Probase: a probabilistic taxonomy for text understanding, in: Proc. of the 2012 ACM SIGMOD Int. Conf. on Management of Data, SIGMOD'12, ACM, ISBN 978-1-4503-1247-9, 2012, pp. 481–492.
[70] H. Yang, C. Rudin, M. Seltzer, Scalable Bayesian rule lists, in: Proceedings of the 34th International Conference on Machine Learning – Volume 70, ICML'17, 2017, pp. 3921–3930, JMLR.org.
[71] J. Zhang, A. Silvescu, V.G. Honavar, Ontology-driven induction of decision trees at multiple levels of abstraction, in: Proc. of the 5th Int. Symposium on Abstraction, Reformulation and Approximation, in: LNCS, vol. 2371, Springer, 2002, pp. 316–323.
[72] Q. Zhang, Y. Yang, H. Ma, Y.N. Wu, Interpreting CNNs via decision trees, in: The IEEE Conference on Computer Vision and Pattern Recognition, CVPR, June 2019.