Explanation as a process: user-centric construction of multi-level and multi-modal explanations [*]

Bettina Finzel, David E. Tafler, Stephan Scheele, and Ute Schmid

Cognitive Systems, University of Bamberg `https://www.uni-bamberg.de/en/cogsys`
`{bettina.finzel,stephan.scheele,ute.schmid}@uni-bamberg.de,`
`david-elias.tafler@stud.uni-bamberg.de`

Abstract. In the last years, XAI research has mainly been concerned with developing new technical approaches to explain deep learning models. Just recent research has started to acknowledge the need to tailor explanations to different contexts and requirements of stakeholders. Explanations must not only suit developers of models, but also domain experts as well as end users. Thus, in order to satisfy different stakeholders, explanation methods need to be combined. While multi-modal explanations have been used to make model predictions more transparent, less research has focused on treating explanation as a process, where users can ask for information according to the level of understanding gained at a certain point in time. Consequently, an opportunity to explore explanations on different levels of abstraction should be provided besides multi-modal explanations. We present a process-based approach that combines multi-level and multi-modal explanations. The user can ask for textual explanations or visualizations through conversational interaction in a drill-down manner. We use Inductive Logic Programming, an interpretable machine learning approach, to learn a comprehensible model. Further, we present an algorithm that creates an explanatory tree for each example for which a classifier decision is to be explained. The explanatory tree can be navigated by the user to get answers of different levels of detail. We provide a proof-of-concept implementation for concepts induced from a semantic net about living beings.

Keywords: Multi-level Explanations · Multi-modal Explanations · Explanatory Processes · Semantic Net · Inductive Logic Programming

1 Introduction

In order to develop artificial intelligence that serves the human user to perform better at tasks, it is crucial to make an intelligent system comprehensible to the human user [20,24]. This requires giving the user an understanding of how an underlying algorithm works (mechanistic understanding) on the one hand and

---

whether the intelligent system fulfills its purpose (functional understanding) on the other hand [26]. Explanations can foster both types of understanding. For example, developers and experts can gain insights into the decision making process and validate the system. Users who have no in depth technical understanding in a field could use explainable systems for training, for instance as implemented in intelligent tutoring systems [27].

Methods to generate explanations are developed in Explainable Artificial Intelligence (XAI) research [10]. A variety of techniques have been proposed, namely approaches that generate post-hoc explanations for deep learned models [1], solutions based on interpretable symbolic approaches [30] as well as a combination of both in hybrid systems [5]. As existing methods are refined and new methods are developed, voices are growing louder about the need for more user-centric solutions (see for example [17,26,30]). Explanations need to fit into the context of the user, meaning the task and level of expertise. However, there is no *one-size-fits-all* explanation method, thus approaches have to be combined.

Current research proposes multi-modal explanations to serve the user's varying need for information [15,8,2], in particular a combination of different explanation strategies (inductive, deductive, contrastive) with explanation modalities (text, image, audio) to represent information accordingly and involve cognitive processes adequately for establishing effective XAI methods. Recent studies show that presenting explanations with different modalities can have a positive influence on the comprehensibility of a decision. Existing approaches combine visual, textual or auditory explanations in multi-modal settings [39,12].

Less focus is given to explanation as a process. Substantial work exists in the area of argumentation, machine learning and explanation [20,38,22,13], where conversations between systems and users follow certain patterns. However, we found that in the current literature not enough attention is given to the fact that functional or mechanistic understanding is developed over a period of time and that users may need different depths and types of information depending on where in the process they are and which level of expertise they have [8]. Template-based explanation approaches that allow humans to drill down into an explanation and to explore its sub-problems in terms of a hierarchical structure have previously been applied to assist ontology engineers in understanding inferences and to correct modelling flaws in formal ontologies [18] as well to justify results from semantic search [29]. Studies have shown that users prefer abstract and simple explanations over complex ones [19], but may ask for more detailed and complex information [40] as well, which should ideally be presented through the best possible and comprehensible explanation strategy. Therefore, involving them in a dialogue, where users can get more detailed explanations if needed, and providing multi-modal explanations at the same time to reveal different aspects of a prediction and its context, seems to be a promising step towards more comprehensible decision-making in human-AI-partnerships.

Our contribution, presented in this paper, is an approach that allows users to understand the classification of a system in a *multi-modal* way and to explore a system's decision step by step through *multi-level* explanations. Multi-modality

is implemented in a way such that explanations can be requested by the user as textual statements and pictures that illustrate concepts and examples from the domain of interest. Multiple levels of explanation are implemented in such a way that the user can pose three types of requests to the system, which then returns explanations with three different levels of detail: a *global* explanation to explain a target class, a *local* explanation to explain the classification of an example with respect to the learned classification model and *drill-down* explanations that reveal more in-depth reasons for the classification of an example. Thus, users can ask for different explanation modalities and details at any point in time in the explanation process in accordance with their understanding of the system's decision that was gained until this point in time.

In a proof-of-concept implementation of our proposed approach, we represent a domain of interest as a semantic net with additional rules to reason about the domain. We train a model based on Inductive Logic Programming (ILP), since it is a method that allows for generating interpretable models [30] for relational, potentially highly complex, domains, where it may be beneficial for the understanding of a user to explain them in a step-by-step manner. In contrast to other interpretable approaches, ILP allows for integrating knowledge and learning relations. It has already been applied to concrete application fields, such as medicine [31] or more abstract examples, like family trees [9], but not yet with a combination of multi-level and multi-modal explanations. In order to generate these explanations, we produce a so-called *explanatory tree* for each example that is to be explained. Each tree is derived from a proof for the classification of an example and extended by references to images that illustrate concepts in the hierarchy. Finally, we allow for interaction in natural language with the explanatory tree through a dialogue interface. Thus, we contribute a concept for multi-level and multi-modal explanations, present an algorithm to construct such explanations and provide a proof-of-concept implementation to realize the explanation process accordingly.

The sections of our paper are organized as follows. In Section 2, we first describe our running example, i.e., the data basis from a relational domain, which consists of a semantic net, reasoning rules and examples. In Section 3, we show how the chosen ILP approach is used to generate an interpretable model by generalizing from the data basis. We proceed with describing our approach to multi-level and multi-modal explanations in Section 4. How to generate an explanatory tree for an example to be explained, is introduced and formalized in Subsection 4.1. The proof-of-concept implementation is presented in Subsection 4.2 and discussed in Subsection 4.3. Finally, we show which research aspects and extensions of the system appear to be of interest for future work in Section 5.

2   A Relational Knowledge Domain

For simple domains, where the model consists just of conjunctions of feature values, it might be sufficient to give a single explanation in one modality. For instance, the prediction that *Aldo plays tennis, if the outlook is sunny and the hu-*

midity is normal (see PlayTennis example from [21]) can be plausibly explained by presenting a verbal statement of the learned constraints. In contrast, if a model is more complex, for instance involving relational structures, multi-level and multi-modal explanations might be helpful to foster understanding.

In the following we introduce a running example that we will refer to throughout the paper to illustrate our approach. We assume the reader is familiar with first-order logic and Prolog (see [35]), but we will restate the key terminology. The main constructs in Prolog are facts, rules and queries. Basic Prolog programs consist of a finite set of Horn clauses, where each is a finite conjunction of literals with at most one positive literal, written in the form
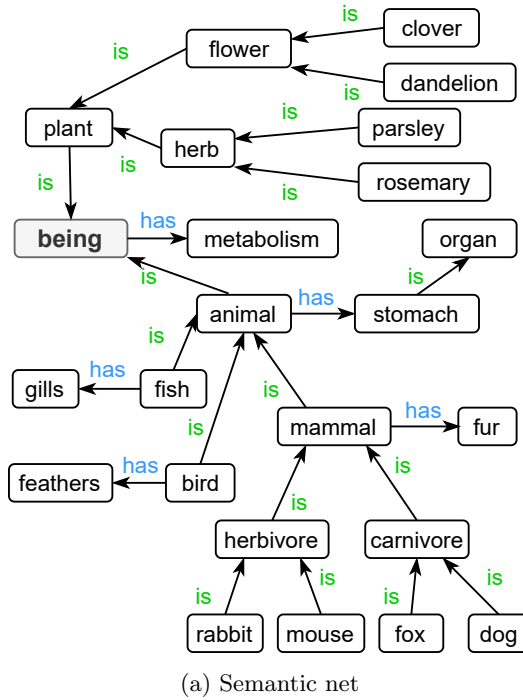
$$A_0 \leftarrow A_1 \wedge A_2 \wedge \ldots \wedge A_n, \text{ where } n \geq 0.$$

Each A_i is an atomic formula of the form $p(t_1, \ldots, t_m)$, consisting of a predicate p and terms t_i, that are either a constant symbol, a variable or a composite expression. The atom A_0 is called the *head* and the conjunction of elements $\bigwedge_{i=1}^{n} A_i$ is called the *body* of a clause. Horn clauses with an empty body are denoted as *facts* and express unconditional statements, otherwise they are called *rules* and express conditional statements. Semantically, we can read a clause as "the conclusion (or *head*) A_0 is true if every A_i in the body is true". Facts are literals with constant terms. Rules express a logical implication to describe that a condition holds if a combination of literals holds, supported by given facts. *Queries* can be used to retrieve information from a Prolog program. Queries can be either facts, to check for their truth or falsity or they can be composite expressions to retrieve terms that make those expressions true. Note that Prolog uses :- to denote ←, "," for conjunction ∧ and every clause ends in a full stop.

Semantic nets are a well-known formalism to represent relational and hierarchical knowledge. They are constructed from a set of nodes and a set of directed, labeled edges, where nodes represent concepts and edges denote the semantic relationships between concepts [6,11]. A semantic network serves as a schema to represent facts in a declarative way and can therefore be implemented, for example in Prolog using predicates to represent relations between concepts.

Fig. 1a represents knowledge about living beings and their relations to each other. The nodes in the semantic net represent concepts and their subset hierarchy is expressed through edges via the `is` relationship, e.g., `birds` belong to class `animal`. The `has` relation denotes properties of a concept, e.g., `bird` *has* `feathers`. Both relations are transitive, e.g., a `carnivore` is an `animal`, because it is a `mammal` which is an `animal`. Fig. 1b shows the corresponding Prolog encoding, where facts consist of a predicate (`is_a` or `has_property`) with one or more *constants* as terms, e.g., `is_a(plant,being)` denotes that plant is a being.

Reasoning over a Prolog program P is based on the inference rule *modus ponens*, i.e., from B and $A \leftarrow B$ one can deduce A, and the first-order resolution principle and *unification* (the reader may refer to [3,35] for more details). For a query q it verifies whether logical variables can be successfully substituted with constants from existing facts or some previously implied conditions from P. That means, an existentially quantified query q is a logical consequence of a program

(a) Semantic net

```
is_a(plant,being).
is_a(animal,being).
is_a(flower,plant).
is_a(clover,flower).
is_a(dandelion,flower).
is_a(herb,plant).
is_a(parsley,herb).
is_a(rosemary,herb).
is_a(fish,animal).
is_a(bird,animal).
is_a(mammal,animal).
is_a(herbivore,mammal).
is_a(carnivore,mammal).
is_a(mouse,herbivore).
is_a(rabbit,herbivore).
is_a(fox,carnivore).
is_a(dog,carnivore).
is_a(stomach,organ).

has_p(being,metabolism).
has_p(animal,stomach).
has_p(fish,gills).
has_p(bird,feathers).
has_p(mammal,fur).
```

(b) Prolog representation

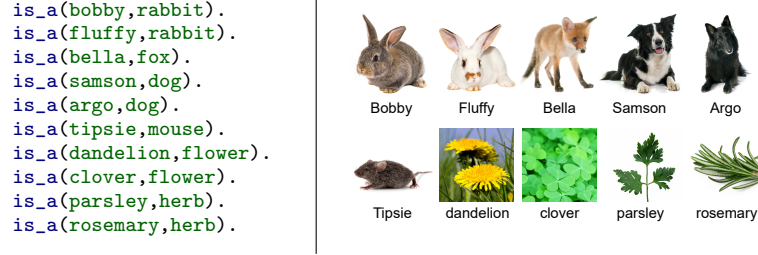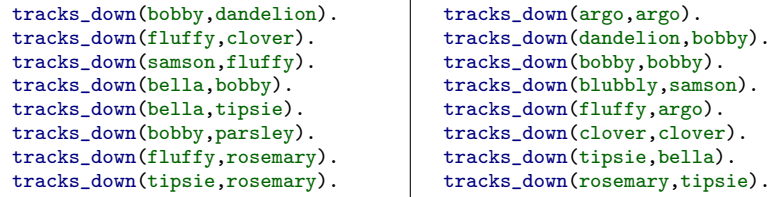Fig. 1: Semantic net of living beings and its representation in Prolog.

P if there is a clause in P with a ground instance $A \leftarrow B_1, ..., B_n, n \geq 0$ such that $B_1, ..., B_n$ are logical consequences of P, and A is an instance of q. Thus, to answer query A, the conjunctive query $\bigwedge_{i=1}^{n} B_i$ is answered first, since A follows from it. For instance, the first reasoning rule from our running example, denoting that `is(A,B)` $\leftarrow$ `is_a(A,B)` can be used to query the semantic net. If we pose the query `is(animal,being)` accordingly, it will be *true*, since the predicate `is_a(animal,being)` is present in the facts. With the second reasoning rule `is(A,B)` (see below) and the query `is(fox,being)` we could, for example, find out that a fox is a living being, due to transitivity.

Additionally, we introduce inheritance and generalization for both relationships, e.g., to express that concept rabbit inherits *has* relationships from mammal. For instance, a `rabbit` *has* `fur`, since it *is* a `mammal` that *has* `fur`. Generalisation, on the other hand, allows us to express properties in a more abstract way, e.g., that an `animal` *has* a `stomach` and since the more general concept of `stomach` *is* `organ`, it follows that `animal` *has* `organ`. Transitivity and inheritance as well as generalisation are expressed by the following reasoning rules:

```
is(A,B)  :-   is_a(A,B).           is(A,B) :-   is_a(A,C), is(C,B).
has(A,X) :-   has_p(A,X).          has(X,Z) :-   has_p(X,Y), has(Y,Z).
has(A,X) :-   is(A,B), has(B,X).   has(A,X) :-   has_p(A,Y), is(Y,X).
```

```
is_a(bobby,rabbit).
is_a(fluffy,rabbit).
is_a(bella,fox).
is_a(samson,dog).
is_a(argo,dog).
is_a(tipsie,mouse).
is_a(dandelion,flower).
is_a(clover,flower).
is_a(parsley,herb).
is_a(rosemary,herb).
```


Fig. 2: Background knowledge T with corresponding example images.

```
tracks_down(bobby,dandelion).        tracks_down(argo,argo).
tracks_down(fluffy,clover).          tracks_down(dandelion,bobby).
tracks_down(samson,fluffy).          tracks_down(bobby,bobby).
tracks_down(bella,bobby).            tracks_down(blubbly,samson).
tracks_down(bella,tipsie).           tracks_down(fluffy,argo).
tracks_down(bobby,parsley).          tracks_down(clover,clover).
tracks_down(fluffy,rosemary).        tracks_down(tipsie,bella).
tracks_down(tipsie,rosemary).        tracks_down(rosemary,tipsie).
```

Fig. 3: Positive (E^+) and negative (E^-) training examples.

We will show later how successful unifications resulting from this step-wise reasoning procedure can be stored in a data structure, which we call *explanatory trees*, to generate explanations with different levels of detail.

The definition of the semantic net and the reasoning rules represent a so-called background theory T [7], about living beings in this case, but it does not include examples yet. In order to represent knowledge about examples, entries for each example can be added to T by including respective predicates such as shown in Figure 2, e.g., to express that `bobby` is an instance of the concept `rabbit`.

Having the background theory complemented by knowledge about concrete examples, which describes relational and hierarchical knowledge, we proceed to take this as input to learn a classification model for a given target class. From a user's perspective, the target class can be understood as a concept, which we want to explain to the user. For our running example, we define a relational classification task, where the model has to learn if and under which conditions a living being would *track down* another living being.

Consider the set of training examples in Figure 3. First, we define a set E^+ of *positive* examples that belong to the target class. For example, the rabbit `bobby` would track down `dandelion`. Secondly, we define a set E^- of *negative* examples that do not belong to the target class, i.e., we define cases that exclude reflexivity, inverse transitivity within species and inverses among all species. For instance, consider the dog `argo` that would not track down itself and obviously also the flower `dandelion` would not track down the rabbit `bobby`.

3   Learning an Interpretable Model with ILP

Inductive Logic Programming (ILP) will be applied to induce a classification model in terms of inferred rules from the given examples and background theory as introduced before. For our running example, the model shall learn whether one living being would track down another living being. ILP is capable of learning a model over hierarchical and non-hierarchical relations that separates positive from negative examples. The basic learning task of ILP (see e.g., [3] and [23]) is described in Algorithm 1 and defined as follows:

An ILP problem is a tuple (T, E^+, E^-) of ground atoms, where the background theory T is a finite set of Horn clauses, and E^+ and E^- are disjoint finite sets of positive and negative example instances. The goal is to construct a first-order clausal theory M we call *model*, which is a set of definite clauses that consistently explains the examples w.r.t. the background theory. Specifically, it must hold that the model M is *complete* and *consistent*, i.e.,

$$\forall p \in E^+ : T \cup M \vDash p, \text{and } \forall n \in E^- : T \cup M \nvDash n,$$

where symbol $\vDash$ denotes the semantic entailment relation and we assume the examples to be noise-free, i.e., this excludes false positives and false negatives.

Algorithm 1: Basic ILP algorithm

Input:
(T, E^+, E^-): an ILP problem
Output:
M: a classification model
Begin:
    $M \leftarrow \varnothing$, initialise the model
    $C \leftarrow \varnothing$, temporary clause
    $Pos \leftarrow E^+$, the set of positive training examples
        While $Pos \neq \varnothing$ **do**
          $C \leftarrow GenerateNewClause(Pos, E^-, T)$
              such that $\exists p \in E^+ : T \cup C \vDash p$
              and $\forall n \in E^- : T \cup C \nvDash n$
              and C is optimal w.r.t. quality criterion $A(C)$
          $Pos \leftarrow Pos \setminus \{p\}$
          $M \leftarrow M \cup \{C\}$
        End
    Return M
End

In particular, it can be seen from Algorithm 1 that it learns in a top-down approach a set of clauses which cover the positive examples while not covering the negative ones. A model is induced by iteratively generating new clauses C using the function $GenerateNewClause$ [9] with T, E^+ and E^-, such that for

```
tracks_down(A,B) :- is(A,carnivore), is(B,herbivore).
tracks_down(A,B) :- is(A,herbivore), is(B,plant).
```

Fig. 4: Learned model

each C there exists a positive example p that is entailed by $C \cup T$, i.e., $C \cup T \vDash p$, while no negative example n is entailed by $C \cup T$. Furthermore, it is of interest to find clauses such that each C is optimal with respect to some *quality criterion* $A(C)$, such as the total number of entailed positive examples. Such *best* clauses are added to model M.

In our approach we use the *Aleph* framework with default settings for the quality criterion [34] to generalize clauses (rules) from (T, E^+, E^-). A well-structured overview on ILP and Aleph in particular is given in Gromowski et al. [9]. Given the examples as well as the background theory for our running example, Aleph learns a model consisting of two rules (see Figure 4). According to these rules, a living being A tracks down another living being B either if A is a carnivore and B is a herbivore, or A is a herbivore and B is a plant.

Note that in our running example, we disregarded carnivorous plants as well as cannibals. Since ILP works based on the *closed world* assumption, these cases do not hold given T.

4   Multi-level and Multi-modal Explanations

Having the model induced from a relational background theory and training examples, we can now proceed to explain the model's prediction for an individual example with our proposed multi-level and multi-modal approach that is presented and discussed in the following subsections. We define and explain how our proposed approach generates explanations and how the user can enter into a dialogue with the system to receive them.

4.1   Explanation Generation

Explanations produced by our approach cover three different levels, which can be described as follows.

- The **first level** reveals a *global* explanation, thus an explanation for the target class (e.g., What does *tracks_down* mean?). This level of detail gives the most shallow explanation for a target class of positive examples.
- The **second level** gives a *local* explanation for the classification of an example by instantiating the global model.
- The **third level** allows for a theoretically endless amount of detail/drill-down. The drill-down follows each clause's body, which can consist of literals that constitute heads of further clauses. Thus producing explanations continues as long as the user asks follow-up questions. However, if the dialogue

reaches a fact, the drill-down ends. The user can then ask for an image in order to receive a visual explanation.

This means, the user can ask for the class of an example, for an explanation for the class decision as well as ask for explanations for the underlying relational concepts and features in a step-by-step manner. We define the terminology and different kinds of explanations more formally in the following paragraphs.

Let $P = M \cup T$ be a Prolog program, which we can query to explain the classification of a positive example $p \in E^+$. We can create a tree ε for each clause in $C \in M$ with $C \vDash p$, which we call an *explanatory tree*. This tree is created as part of performing a proof for example p, given a clause $C \in M$ and the corresponding background theory T such as introduced in [3]. We extend the proof computation by storing ε consisting of successful query and sub-query unifications. We check, whether an example $p \in E^+$ can be proven to be entailed by $T \cup C$, where $C \in M$, all generated by Algorithm 1.

The tree can be traversed to yield explanations of different levels of detail for our proposed dialogue approach. In order to generate a *global* explanation for a target class c (see Def. 1), the set of all clauses from model M of that target class are presented. In order to explain a positive example p globally, only the clauses from M that entail p are presented. The global explanation corresponds to the body of each clause C from M, while the target class is encoded in the heads of all clauses in M. In order to generate a *local* explanation (see Def. 2), a ground clause $C\theta$ has to be found for C taken from M or T, under substitution θ of the terms in C. A local explanation is derived from a successful proof of q initialized to the head of $C\theta$, where the body of the clause $C\theta$ is entailed by M and T. The *drill-down* is applied to a local explanation $C\theta$ (see Def. 3), and is defined as the creation of a subsequent local explanation for some ground literal $B_i\theta$ from the set of all literals from the body of $C\theta$, given that the body of $C\theta$ is not empty. If the head of $C\theta$ is a fact and consequently the body of $C\theta$ is empty, the drill-down stops.

Definition 1 (Global explanation). *A* global *explanation for a target class c is given by the set M of all clauses in the learned model.*

Definition 2 (Local explanation). *A* local *explanation for a query q is a ground clause $C\theta$ where $C \in M \cup T$ such that $q = head(C\theta)$ and $M \cup T \vDash body(C\theta)$.*

Definition 3 (Drill down of a local explanation). *A* drill down *for a local explanation $C\theta$ is given by some literal in $body(C\theta) = B_1, B_2, \ldots, B_n$ where $n \geq 0$ such that either $head(C\theta)$ is a fact, i.e., $head(C\theta) \vdash true$ $(body(C\theta) = \varnothing)$; or otherwise $body(C\theta) \neq \varnothing$ and we create a local explanation for some $B_i\theta$, where $1 > i \leq n$.*

Accordingly, the explanatory tree ε is constructed such that, for each successful unification of a query q, ground q is the parent node of ground clauses resulting from the proof of q. The explanatory tree ε can be traversed up and down in a dialogue to get explanations at all defined levels of detail in the form of natural language expressions or images at leaf nodes.

4.2   Explanatory Dialogue

As presented above, our multi-level and multi-modal explanation approach allows the user to enter into a dialogue with the system and ask for explanations on three different levels. Accordingly, users can pose various types of questions, depending on the need for information and detail.

The input, internal program structure as well as the output of an ILP program is represented in expressive first-order predicate logic, in Prolog. Nevertheless, although its output is readable for humans, it is not necessarily comprehensible, due to factors like the complexity of the domain of interest and the unusual syntax. We therefore realize the explanatory dialogue by generating natural language expressions.

Template-based generation is commonly used to generate natural language explanations for logic-based models, e.g., see Siebers et al. [32] and Musto et al. [25] or [18]. Our template consists of domain-specific as well as domain-independent transformation rules. For example the *has_a* and *is_a* relations can be easily translated for any kind of data set, which is structured as a semantic net. For binary predicates we connect the first argument with the second argument through a transformed version of the predicate. In case of a n-ary predicate, arguments are added by *and*. Each parent node from the explanatory tree is connected to its direct children through the word *because*, since giving a reason is at the core of implication. Consider the first rule of our learned model, presented in section 3. Transforming this rule to formulate for example a *global* explanation in natural language results in the sentence:"A tracks down B, because A is a carnivore and B is a herbivore.". For a *local* explanation, the sentence is, e.g.:"Bella tracks down Bobby, because Bella is a carnivore and Bobby is a herbivore.". Beyond this template-based expression generation, our dialogue provides some static content, such as introductory content, advice and an epilogue, when the user quits the program.

An example of an explanatory dialogue for the classification of the relationship between *Bobby* and *dandelion* is presented in Figure 5. The different levels of explanations correspond to the levels of the previously generated explanatory tree. The user can explore the whole tree, following the paths up and down, depending on the type of request (see Figure 5). Due to space restrictions we are only presenting the dialogue for local explanations and drill-down requests. The user could ask for a global explanation by posing the question:"What does tracks_down mean?". The system would return an answer based on the verbalization of the predicate `is` and the constants `carnivore` and `herbivore` to explain the first rule of the learned model from Section 3.

4.3   Proof-of-Concept Implementation

The dialogue presented in Fig. 4.2 illustrates the proof-of-concept implementation of our approach. The code is accessible via gitlab[1]. In general, the approach

---

[1] Gitlab repository of the proof-of-concept implementation: `https://gitlab.rz.` `uni-bamberg.de/cogsys/public/multi-level-multi-modal-explanation`
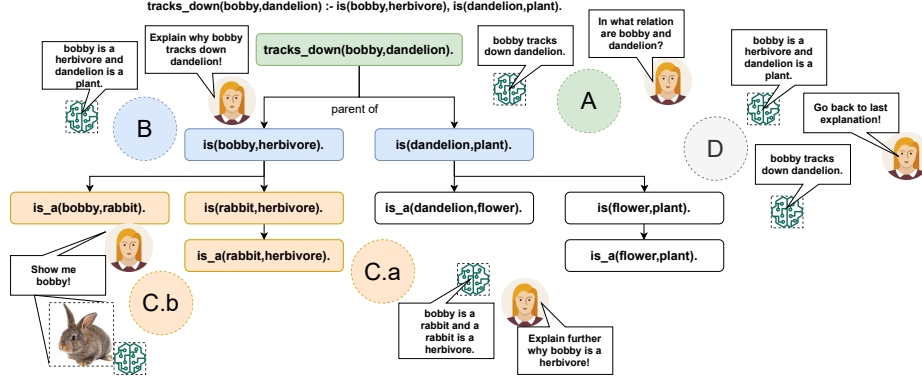
Fig. 5: An explanatory tree for `tracks_down(bobby,dandelion)`, that can be queried by the user to get a *local* explanation why Bobby tracks down dandelion (steps A and B). A dialogue is realized by different *drill-down* questions, either to get more detailed verbal explanations or visual explanations (steps C.a and C.b)). Furthermore, the user can return to the last explanation (step D).

is not only applicable to the domain presented here. Any Prolog program including a model learned with ILP can be explained with our approach. The only precondition is, that the template-based verbalization of the nodes from the explanatory tree can be performed by the domain-independent transformation rules. Otherwise, domain-specific transformation rules must be defined first. Furthermore, we want to point out that the introduced algorithm fulfills correctness, since it is complete and consistent with respect to the ILP problem solution. Finally, first empirical investigations show that humans perform better in decision-making and trust more into the decisions of an intelligent system, if they are presented with visual as well as verbal explanations [37].

5   Conclusion and Outlook

Explanations of decisions made by intelligent systems need to be tailored to the needs of different stakeholders. At the same time there exists no *one-size-fits-all* explanation method. Consequently, an approach that combines explanation modalities and that provides explanation as a step-by-step process, is promising to satisfy various users. We presented an approach that combines textual and visual explanations, such that the user can explore different kinds of explanations by posing requests to the system through a dialogue. We introduced an algorithm as well as our proof-of-concept implementation of it.

In comparison to other explainable state-of-the-art systems that present explanations rather as static content [1], our approach allows for step-by-step exploration of the reasons behind a decision of an intelligent system. Since our approach is interpretable, it could help users in the future to uncover causal-

ities between data and a system's prediction. This is especially important in decision-critical areas, such as medicine [14,4,31].

Other recent interactive systems enable the user to perform corrections on labels and to act upon wrong explanations, such as implemented in the CAIPI approach [36], they allow for re-weighting of features for explanatory debugging, like the EluciDebug system [16] and correcting generated verbal explanations and the underlying model through user-defined constraints, such as implemented in the medical-decision support system LearnWithME [31]. Our approach could be extended by correction capabilities in the future, in addition to requesting information from the system to better understand its operation or purpose. In that way, explanations would be bi-directional.

We presented a proof-of-concept implementation of our approach that could be technically extended in the future by combining explanations with linked data, e.g., to integrate formal knowledge from ontologies combined with media from open data repositories, which would allow for more flexibility in the presentation of content based on semantic search[2]. Furthermore, we envisage to explain decisions of deep neural networks using ILP, as presented in [28].

In future work, we aim to systematically evaluate our approach with empirical user studies, recognizing design dimensions of XAI put forth by Sperrle et al. [33]. In this context, our multi-modal and multi-level approach allows for precise control and manipulation of experimental conditions. For instance, we are currently preparing an empirical study to evaluate the effectiveness of the combination of multi-level and multi-modal explanations with respect to user performance and understanding of a model. It is planned to apply the proposed approach to a decision-critical scenario, in particular to asses pain and emotions in human facial expressions for a medical use case.

In order to meet a user-centric explanation design and to close the semantic gap between data, model and explanation representation, the use of hybrid approaches seems promising as a future step. Especially in the field of image-based classification, it is an exciting task to consider sub-symbolic neural networks as sensory components and to combine them with symbolic approaches as drivers of knowledge integration. In this way, models can be validated and corrected with the help of expert knowledge. Furthermore, hybrid approaches allow for modeling the context of an application domain in a more expressive and complex manner and, in addition, taking into account users' context and individual differences. Our approach is a first step towards this vision.

References

1. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable artifi-

---

[2] Retrieval through semantic search can be performed for example over semantic *mediawiki*: https://www.semantic-mediawiki.org/wiki/Help:Semantic_search

cial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion **58**, 82–115 (2020)

2. Baniecki, H., Biecek, P.: The grammar of interactive explanatory model analysis. CoRR **abs/2005.00497** (2020)

3. Bratko, I.: Prolog Programming for Artificial Intelligence. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1986)

4. Bruckert, S., Finzel, B., Schmid, U.: The next generation of medical decision support: A roadmap toward transparent expert companions. Frontiers in Artificial Intelligence **3**, 75 (2020)

5. Calegari, R., Ciatto, G., Omicini, A.: On the integration of symbolic and sub-symbolic techniques for XAI: A survey. Intelligenza Artificiale **14**(1), 7–32 (2020)

6. Chein, M., Mugnier, M.L.: Graph-based knowledge representation: computational foundations of conceptual graphs. Springer Science & Business Media (2008)

7. De Raedt, L., Lavrač, N.: The many faces of inductive logic programming. In: Komorowski, J., Raś, Z.W. (eds.) Methodologies for Intelligent Systems. pp. 435–449. Lecture Notes in Computer Science, Springer Berlin Heidelberg (1993)

8. El-Assady, M., Jentner, W., Kehlbeck, R., Schlegel, U., Sevastjanova, R., Sperrle, F., Spinner, T., Keim, D.: Towards XAI: Structuring the Processes of Explanations. In: Proceedings of the ACM CHI Conference Workshop on Human-Centered Machine Learning Perspectives at CHI'19. p. 13 (2019)

9. Gromowski, M., Siebers, M., Schmid, U.: A process framework for inducing and explaining datalog theories. ADAC **14**(4), 821–835 (2020)

10. Gunning, D., Aha, D.: DARPA's Explainable Artificial Intelligence (XAI) Program. AI Magazine **40**(2), 44–58 (2019)

11. Hartley, R.T., Barnden, J.A.: Semantic networks: visualizations of knowledge. Trends in Cognitive Sciences **1**(5), 169–175 (1997)

12. Hendricks, L.A., Hu, R., Darrell, T., Akata, Z.: Grounding visual explanations. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)

13. Hilton, D.J.: A Conversational Model of Causal Explanation. European Review of Social Psychology **2**(1), 51–81 (1991)

14. Holzinger, A., Langs, G., Denk, H., Zatloukal, K., Müller, H.: Causability and explainability of artificial intelligence in medicine. WIREs Data Mining and Knowledge Discovery **9**(4) (2019)

15. Holzinger, A., Malle, B., Saranti, A., Pfeifer, B.: Towards multi-modal causability with Graph Neural Networks enabling information fusion for explainable AI. Information Fusion **71**, 28–37 (2021)

16. Kulesza, T., Stumpf, S., Burnett, M., Wong, W.K., Riche, Y., Moore, T., Oberst, I., Shinsel, A., McIntosh, K.: Explanatory debugging: Supporting end-user debugging of machine-learned programs. In: 2010 IEEE Symposium on Visual Languages and Human-Centric Computing. pp. 41–48. IEEE (2010)

17. Langer, M., Oster, D., Speith, T., Heckrmanns, H., Kästner, L., Schmidt, E., Sesing, A., Baum, K.: What Do We Want From Explainable Artificial Intelligence (XAI)? – A Stakeholder Perspective on XAI and a Conceptual Model Guiding Interdisciplinary XAI Research. Artificial Intelligence p. 103473 (2021)

18. Liebig, T., Scheele, S.: Explaining entailments and patching modelling flaws. Künstliche Intell. **22**(2), 25–27 (2008)

19. Lombrozo, T.: Simplicity and probability in causal explanation. Cognitive Psychology **55**(3), 232–257 (2007)

20. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial intelligence **267**, 1–38 (2019)

21. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
22. Možina, M., Žabkar, J., Bratko, I.: Argument based machine learning. Artificial Intelligence **171**(10), 922–937 (2007)
23. Muggleton, S., de Raedt, L.: Inductive Logic Programming: Theory and methods. The Journal of Logic Programming **19-20**, 629–679 (1994)
24. Muggleton, S.H., Schmid, U., Zeller, C., Tamaddoni-Nezhad, A., Besold, T.: Ultra-Strong Machine Learning: comprehensibility of programs learned with ILP. Machine Learning **107**(7), 1119–1140 (2018)
25. Musto, C., Narducci, F., Lops, P., De Gemmis, M., Semeraro, G.: ExpLOD: A Framework for Explaining Recommendations based on the Linked Open Data Cloud. In: Proceedings of the 10th ACM Conference on Recommender Systems. pp. 151–154. ACM, Boston Massachusetts USA (2016)
26. Páez, A.: The Pragmatic Turn in Explainable Artificial Intelligence (XAI). Minds and Machines **29**(3), 441–459 (2019)
27. Putnam, V., Conati, C.: Exploring the Need for Explainable Artificial Intelligence (XAI) in Intelligent Tutoring Systems (ITS). Los Angeles p. 7 (2019)
28. Rabold, J., Siebers, M., Schmid, U.: Explaining black-box classifiers with ILP – empowering LIME with Aleph to approximate non-linear decisions with relational rules. In: Proc. Int. Conf. Inductive Logic Programming. pp. 105–117. Lecture Notes in Computer Science, Springer International Publishing (2018)
29. Roth-Berghofer, T., Forcher, B.: Improving understandability of semantic search explanations. Int. J. Knowl. Eng. Data Min. **1**(3), 216–234 (2011)
30. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. **1**(5) (2019)
31. Schmid, U., Finzel, B.: Mutual explanations for cooperative decision making in medicine. KI-Künstliche Intelligenz pp. 1–7 (2020)
32. Siebers, M., Schmid, U.: Please delete that! Why should I? KI - Künstliche Intelligenz **33**(1), 35–44 (2019)
33. Sperrle, F., El-Assady, M., Guo, G., Chau, D.H., Endert, A., Keim, D.: Should We Trust (X)AI? Design Dimensions for Structured Experimental Evaluations. arXiv:2009.06433 [cs] (2020), arXiv: 2009.06433
34. Srinivasan, A.: The Aleph Manual. `http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/`
35. Sterling, L., Shapiro, E.: The Art of Prolog: Advanced Programming Techniques. MIT Press, Cambridge, MA, USA (1986)
36. Teso, S., Kersting, K.: Explanatory interactive machine learning. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. pp. 239–245 (2019)
37. Thaler, A., Schmid, U.: Explaining machine learned relational concepts in visual domains – effects of perceived accuracy on joint performance and trust. In: Proceedings of the 43rd Annual Conference of the Cognitive Science Society (CogSci'21, Vienna). Cognitive Science Society ((to appear))
38. Walton, D.: A Dialogue System for Evaluating Explanations. In: Walton, D. (ed.) Argument Evaluation and Evidence, pp. 69–116. Law, Governance and Technology Series, Springer International Publishing, Cham (2016)
39. Weitz, K., Schiller, D., Schlagowski, R., Huber, T., André, E.: "Let me explain!": exploring the potential of virtual agents in explainable AI interaction design. Journal on Multimodal User Interfaces (2020)
40. Zemla, J.C., Sloman, S., Bechlivanidis, C., Lagnado, D.A.: Evaluating everyday explanations. Psychonomic Bulletin & Review **24**(5), 1488–1500 (2017)