Explaining $\mathcal{ELH}$ Concept Descriptions through Counterfactual Reasoning

Leonie Nora Sieger
Data Science Research Group
Paderborn University, Germany
leonie.nora.sieger@uni-paderborn.de

Stefan Heindorf
Data Science Research Group
Paderborn University, Germany
heindorf@uni-paderborn.de

Yasir Mahmood
Data Science Research Group
Paderborn University, Germany
yasir.mahmood@uni-paderborn.de

Lukas Blübaum
Data Science Research Group
Paderborn University, Germany
lukasbl@campus.uni-paderborn.de

Axel-Cyrille Ngonga Ngomo
Data Science Research Group
Paderborn University, Germany
axel.ngonga@upb.de

ABSTRACT

Knowledge bases are widely used for information management, enabling high-impact applications such as web search, question answering, and natural language processing. They also serve as the backbone for automatic decision systems, e.g., for medical diagnostics and credit scoring. As stakeholders affected by these decisions would like to understand their situation and verify how fair the decisions are, a number of explanation approaches have been proposed. An intrinsically transparent way to do classification is by using concepts in description logics. However, these concepts can become long and difficult to fathom for non-experts, even when verbalized. One solution is to employ counterfactuals to answer the question, "How must feature values be changed to obtain a different classification?" By focusing on the minimal feature changes, the explanations are short, human-friendly, and provide a clear path of action regarding the change in prediction. While previous work investigated counterfactuals for tabular data, in this paper, we transfer the notion of counterfactuals to knowledge bases and the description logic $\mathcal{ELH}$. Our approach starts by generating counterfactual candidates from concepts, followed by selecting the candidates requiring the fewest feature changes as counterfactuals. When multiple counterfactuals exist, we rank them based on the likeliness of their feature combinations. We evaluate our method by conducting a user survey to determine which counterfactual candidates participants prefer for explanation.

KEYWORDS

XAI, machine learning, description logic, counterfactuals

K-CAP '23, December 5 - 7, 2023, Pensacola, Florida, USA
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Leonie Nora Sieger, Stefan Heindorf, Yasir Mahmood, Lukas Blübaum, and Axel-Cyrille Ngonga Ngomo. 2023. Explaining $\mathcal{ELH}$ Concept Descriptions through Counterfactual Reasoning. In *Proceedings of Twelfth International Conference on Knowledge Capture, K-CAP 2023 (K-CAP '23).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Knowledge bases (KBs) are commonly used to represent information in various domains. KBs such as DBpedia [3], Wikidata [53], or YAGO [51] are used in web applications, including information retrieval [47], information generation [44], web search [13] and question answering [27]. In the medical domain, KBs such as DrugBank [56], Snomed [14], and STRING [52] are widely used for predicting whether a molecule is safe, whether it helps against a certain disease, and what the functions of proteins are [32]. Further applications include medical diagnostics [17], credit scoring [58] and hiring decisions [19]. Typical machine learning tasks include predicting whether the information about an entity is complete [25, 39], correct [18], and what category an entity falls in [26, 57]. In many of these domains, explaining algorithmic decisions of AI systems to stakeholders is important [1, 2, 28, 42]: (i) subjects affected by model decisions would like to understand their situation and verify fair decisions; (ii) data scientists would like to debug and improve their model; (iii) regulatory entities would like to check compliance with laws and regulations.

For KBs, concepts in description logics (DLs) can serve as transparent, white-box models for binary classification and many approaches for learning concepts from positive and negative examples have been proposed [15, 22, 30, 37, 38]. For each individual in the KB, they predict whether a concept holds, i.e., whether the individual should be classified as a positive instance for the concept. However, the learned concepts can become long and complex and often exceed 20 tokens, sometimes as many as 1,000 [30]. This jeopardizes their use as short, human-friendly explanations. Moreover, as a concept provides an explanation *for each* individual (global explanation), it often contains many parts that are irrelevant to explain the prediction of a *single individual* (local explanation). To mitigate these issues, counterfactual explanations (CEs) can serve as a form of short and actionable explanations [43]. CEs focus on an antecedent that would have caused a different outcome (classification) had it been the case [50]. CEs answer the question of how

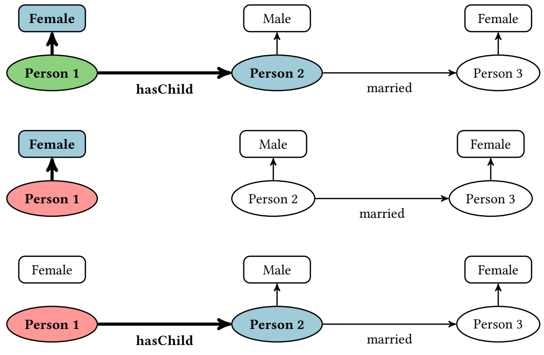

Figure 1: The concept Female ⊓ ∃hasChild.⊤ classifies Person 1 as a mother in KB 1 (green), while its corresponding counterfactuals in KBs 2 and 3 (red) are not classified as such.

the classifier's input needs to be minimally changed to arrive at a different prediction [54]. Dandl et al. [20] generalize this idea and take further criteria into account. They propose a multi-objective optimization problem with four objectives: (1) the prediction for the CF should be as close as possible to the desired prediction; (2) the CF should be as similar as possible to the original instance; (3) feature changes should be sparse; (4) the CF should have plausible/likely feature values/combinations.

While CEs are widely used for tabular data [20, 50, 54], in this paper we transfer the notion of CEs to DLs and generate simple explanations for *individuals*. As a first step in this direction, we focus on the DL $\mathcal{ELH}$ that is designed to balance expressivity and computational tractability by limiting the language constructs and supporting efficient reasoning algorithms. Given an $\mathcal{ELH}$-KB $\mathcal{K}$, a concept C, and an individual x such that $C(x)$ holds (respectively does not hold) in $\mathcal{K}$, we generate the counterfactual candidates $\mathcal{K}'$ from $\mathcal{K}$ in which $C(x)$ does not hold (holds) by applying changes only to the ABox. We also define criteria of non-redundancy, locality and minimality with the goal of generating practically useful CEs for ML models focusing on the individual x. Figure 1 shows an example with two counterfactual candidates. In line with previous works [20, 54], we define counterfactuals (CFs) as those candidates which are most similar to the original KB. Finally, we rank these CFs according to the plausibility with which they appear in the real world, i.e., the *likeliness* of the combination of their features, to construct CEs from the most plausible ones. We also conducted a user survey to investigate if the selected CEs are indeed preferred by users. To summarize, our contributions are as follows:

(1) We formally define CFs and CEs for individuals in DLs.
(2) We provide an algorithm to compute CEs for concept descriptions in $\mathcal{ELH}$.
(3) We provide a heuristic (likeliness) to choose among multiple counterfactuals such that the user will only see a small number of counterfactual explanations.
(4) We evaluate our CEs and likeliness measures via a user study.

In what follows, Section 2 discusses related work, Section 3 introduces preliminaries, Section 4 formalizes the notion of CEs in $\mathcal{ELH}$ and presents our algorithms to generate CEs, and Section 5 presents the user study to investigate user preferences. Finally, Section 6 discusses our results. For all data, code and materials needed for reproducing, and a full version including proof details, see our repository [1].

2 RELATED WORK

Current explanation approaches for description logics include justification, axiom pinpointing, as well as abduction (for both ABoxes and TBoxes). *Justification*-based approaches [31, 34, 35] aim at finding explanations of why a TBox axiom (also called general concept inclusions or subsumption) is entailed by a knowledge base. In contrast, our work focuses on the entailment of ABox axioms. The explanations, also called justifications, can be computed via axiom pinpointing [10, 11, 49], which yields a (minimal) subset of TBox axioms that entails a given TBox axiom. *Abductive* reasoning computes a hypothesis that, together with the knowledge base, is sufficient to entail a given set of axioms. ABox abduction [21, 36, 46] expresses both the hypothesis and observation in terms of ABox axioms; TBox abduction [23, 29] expresses them in terms of TBox axioms. The absence of negation in $\mathcal{ELH}$ implies that ABox abduction and CEs (for adding an assertion) are closely related: if an ABox axiom does not hold in a KB ($\mathcal{K}$), then the candidate hypotheses (H) in abduction correspond to counterfactual candidates (updated knowledge base $\mathcal{K} \cup H$). Nevertheless, the picture is different for more expressive DLs involving negation (see Section 6). Moreover, the goals, applications, and optimization problems are completely different. The goal of abduction is to explain why a certain observation does not hold whereas the goal of our counterfactuals is to explain the white-box machine learning model (=concept) learned by a concept learner. The difference becomes particularly apparent when we decide which counterfactual candidate is considered as an actual counterfactual. A typical abduction approach searches for the minimal hypothesis among all potential hypothesis. Our counterfactual approach takes further criteria into account when searching for a counterfactual—e.g., likeliness and locality.

Once abductive explanations or justifications have been obtained, ontology *repairing* [6–8] provides means to make minimal changes to a given knowledge base, enforcing or preventing a particular entailment. Moreover, approaches that employ ABox repairing [12] address the task of making an inconsistent ABox consistent before answering queries on the knowledge base. While repairing approaches aim to make minimal changes to a knowledge base, our definition of optimal changes to a knowledge base is different and we take further criteria into account (e.g., our likelihood and locality measures). Whereas Lehmann and Turhan [41]developed a framework for measuring the similarity of *concepts*, our setting requires measuring the similarity of *individuals* (rather than concepts).

Although counterfactuals have been mentioned in the context of description logics before, their definitions are vastly different from ours. Filandrianos et al. [24] recently proposed a framework for computing counterfactual explanations for *black-box* classifiers whereas we aim to explain *white-box* models—namely concepts obtained from concept learners [15, 22, 30, 37, 38, 48]. Iannone et al. [33] use the term "counterfactual" for negated residuals (i.e., parts of a *concept*) and use them to prune the search space of a

[1] https://github.com/LNSieger/Counterfactual-Explanations-DL-ELH/

concept learner. In contrast, our counterfactuals are (individuals in) knowledge bases and we use them to explain the output of a concept learner.

3 PRELIMINARIES

We give a brief overview of the DL $\mathcal{ELH}$. For further details, we refer the reader to existing literature [5, 14].

Table 1: $\mathcal{ELH}$ description logic constructs.

	Syntax	Semantics	Construct
Concepts	$\top$	$\Delta^{\mathcal{I}}$	top concept
	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$	atomic concepts
	r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	atomic roles
	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	intersection
	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}$ with $(x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$	existential restriction
ABox	$C(x)$	$x^{\mathcal{I}} \in C^{\mathcal{I}}$	concept assertion
	$r(x, y)$	$(x^{\mathcal{I}}, y^{\mathcal{I}}) \in r^{\mathcal{I}}$	role assertion
TBox	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	concept subsumption
	$r \sqsubseteq s$	$r^{I} \subseteq s^{\mathcal{I}}$	role subsumption

The Description Logic $\mathcal{ELH}$. In DLs [5], knowledge is represented by concept descriptions built from atomic concepts $A, B \in N_C$ and roles $r, s \in N_R$, where N_C and N_R are finite sets of concept and role names. Every concept name A as well as the top concept $\top$ is a concept description. As depicted in Table 1, other $\mathcal{ELH}$-concept descriptions are constructed using the intersections ($C \sqcap D$) and existential restrictions ($\exists r.C$) from the concept descriptions C, D and role r. Their semantics is defined in terms of an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ which consists of the non-empty set $\Delta^{\mathcal{I}}$, called interpretation domain, and the function $\cdot^{\mathcal{I}}$, called interpretation function, that assigns each $A \in N_C$ a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each $r \in N_R$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ [40]. Furthermore, each individual $x \in N_I$, where N_I is a finite set of individual names, is assigned an element $x^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ [40].

Let C, D be concepts, r, s be role names, and $x, y \in N_I$ be individuals. A *TBox* is a set of concept inclusion ($C \sqsubseteq D$) and role inclusion axioms ($r \sqsubseteq s$). A concept (resp., role) assertion is an expression of the form $C(x)$ ($r(x, y)$). An *ABox* is a set of concept and role assertion axioms. Finally, a knowldge base $\mathcal{K}$ consists of a TBox and an ABox. An interpretation $\mathcal{I}$ is a model of the KB $\mathcal{K}$ (denoted as $\mathcal{I} \models \mathcal{K}$) iff $\mathcal{I}$ satisfies all axioms in the TBox and ABox. An individual $x \in N_I$ is an instance of a concept C with respect to $\mathcal{K}$, written $\mathcal{K} \models C(x)$ iff in all models $\mathcal{I}$ of $\mathcal{K}$, we have that $a^{\mathcal{I}} \in C^{\mathcal{I}}$. We say that C holds for x in $\mathcal{K}$ if $\mathcal{K} \models C(x)$.

In this paper, we make the unique-name assumption (UNA) and require that $x^{\mathcal{I}} \neq y^{\mathcal{I}}$ for individuals x, y such that $x \neq y$ [4, 5].

4 COUNTERFACTUALS IN $\mathcal{ELH}$

Following Wachter et al. [54] and Dandl et al. [20], who defined CFs for black-box machine learning models with fixed-size input vectors, we transfer their definition to concept assertions in DL. Given a KB $\mathcal{K}$, a concept $C \in N_C$ and an individual $x \in N_I$, then a KB $\mathcal{K}'$ is a counterfactual candidate for $C(x)$ iff either $\mathcal{K} \models C(x)$ and $\mathcal{K}' \not\models C(x)$, or $\mathcal{K} \not\models C(x)$ and $\mathcal{K}' \models C(x)$. That is, the evaluation of $C(x)$ differs with respect to $\mathcal{K}$ and $\mathcal{K}'$ where $\mathcal{K}$ and $\mathcal{K}'$ are defined over the same atomic concepts and roles.

A counterfactual candidate can be seen as an intermediate step in our approach to creating CFs. We formalize counterfactuals as a response to a user's request to change an ABox (i.e., requesting a different scenario). Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB, and $C(x)$ be a concept assertion for an individual x in $\mathcal{K}$. We call the pair $P = \langle C(x), U \rangle$ a *counterfactual request*, where $U \in \{\text{add}, \text{rem}\}$. Let $\mathcal{K}' = (\mathcal{T}, \mathcal{A}')$ be a KB such that: $\mathcal{K}' \models C(x)$ if $U = \text{add}$, or $\mathcal{K}' \not\models C(x)$ if $U = \text{rem}$. Then we say that $\mathcal{K}'$ *fulfills* the request P and denote this by $\mathcal{K}' \vdash P$. Furthermore, we call $\mathcal{K}'$ a counterfactual candidate for P in $\mathcal{K}$. We assume that $\mathcal{K}$ does not already fulfill P and hence needs to be appropriately updated. The fact that $\mathcal{K} \nvdash P$ and $\mathcal{K}' \vdash P$ corresponds to the first objective (closeness to desired prediction) of CEs proposed by Dandl et al. [20]. The creation of a counterfactual candidate can also be denoted as an update $\mathcal{K} \longrightarrow \mathcal{K}'$. To update a KB (when $U = rem$), it is often necessary to change multiple assertions from which the target assertion can be inferred. Due to the complex nature of TBoxes, there are sometimes multiple ways to achieve this.

Example 4.1. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T} := \{B \sqcap C \sqsubseteq D\}$ and $\mathcal{A} := \{B(x), C(x), D(x)\}$. Moreover let $P = \langle D(x), rem \rangle$ be a counterfactual request. Then there are three updates, $\mathcal{K}_1' := (\mathcal{T}, \{B(x)\})$, $\mathcal{K}_2' := (\mathcal{T}, \{C(x)\})$ and $\mathcal{K}_3' := (\mathcal{T}, \emptyset)$ that fulfill P.

Clearly, candidates *without redundancy* having *minimal changes* are preferred in such scenarios. Intuitively, an update $\mathcal{K} \longrightarrow \mathcal{K}'$ is non-redundant if no changes were made in $\mathcal{K}'$ so that avoiding these changes would still fulfill the counterfactual request. In the following, $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and $\mathcal{K}' = (\mathcal{T}, \mathcal{A}')$ denote two KBs with the same TBox ($\mathcal{T}$) and different ABoxes ($\mathcal{A}$ and $\mathcal{A}'$, respectively) unless stated otherwise. We write $\Delta(\mathcal{A}, \mathcal{A}')$ to denote the symmetric difference of $\mathcal{A}$ and $\mathcal{A}'$ defined as $\Delta(\mathcal{A}, \mathcal{A}') := (\mathcal{A} \backslash \mathcal{A}') \cup (\mathcal{A}' \backslash \mathcal{A})$.

Definition 4.2 (Non-redundancy of Changes). Let $\mathcal{K}$ and $\mathcal{K}'$ be two KBs and $P = \langle C(x), U \rangle$ be a CF request. If $\nexists D(y) \in \mathcal{A} \setminus \mathcal{A}'$ such that $(\mathcal{T}, \mathcal{A}' \cup \{D(y)\}) \vdash P$ if $U = rem$, and $\nexists D(y) \in \mathcal{A}' \setminus \mathcal{A}$ so that $(\mathcal{T}, \mathcal{A}' \setminus \{D(y)\}) \vdash P$ if $U = add$, then the update $\mathcal{K} \longrightarrow \mathcal{K}'$ is non-redundant.

In Example 4.1, the updates $\mathcal{K} \longrightarrow \mathcal{K}_1'$ and $\mathcal{K} \longrightarrow \mathcal{K}_2'$ are non-redundant, whereas the update $\mathcal{K} \longrightarrow \mathcal{K}_3'$ is redundant.

When creating a CE with respect to $C(x)$ for an individual x, it might not be desired to affect concept assertions for other individuals $y \neq x$. As a result, CE creation should restrict the allowed changes to those concepts D such that $D(x) \in \mathcal{A}$. This way, CEs answer the question of what features of x should be modified to change its classification.

Definition 4.3 (Local counterfactual candidates). Let $\mathcal{K}$ and $\mathcal{K}'$ be two KBs and P be a CF request. If $\nexists D(y) \in \Delta(\mathcal{A}, \mathcal{A}')$ where $y \neq x$, then $\mathcal{K}'$ is a local counterfactual candidate for P in $\mathcal{K}$.

As proposed by Dandl et al. [20], a CF is a counterfactual candidate with minimum feature changes. In a KB $\mathcal{K}$, we operationalize this via the notion of edit distance, i.e., the number of additions or removals of axioms necessary to create the CF $\mathcal{K}'$ from $\mathcal{K}$ such that $\mathcal{K}'$ fulfils a request P. We will only allow to update an ABox and define the edit distance with respect to changes in the ABox.

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and $\mathcal{K}' = (\mathcal{T}, \mathcal{A}')$ be two KBs. We let $A^x_{\mathcal{K}}$ (resp., $A^x_{\mathcal{K}'}$) denote the set of all concept (D) and role names (r) in $\mathcal{K}$ ($\mathcal{K}'$), such that $D(x) \in \mathcal{A}$ or $r(x, y) \in \mathcal{A}$ (resp., in $\mathcal{A}'$). Then the counterfactuals for $P = \langle C(x), U\rangle$ in $\mathcal{K}$ are the KBs $\mathcal{K}'$ such that $\mathcal{K}' \vdash P$ and $\mathcal{K}'$ have the lowest *edit distance* to $\mathcal{K}$. The edit distance $\delta_{cf}(\mathcal{K}, \mathcal{K}')$ of a CF is formalized via the symmetric difference $\Delta(A^x_{\mathcal{K}}, A^x_{\mathcal{K}'})$ as follows.

$$\delta_{cf}(\mathcal{K}, \mathcal{K}') = |\Delta(A^x_{\mathcal{K}}, A^x_{\mathcal{K}'})| \quad (1)$$

Definition 4.4 (Minimal Changes). Let $\mathcal{K}$ and $\mathcal{K}'$ be two KBs and P be a CF request. Then, $\mathcal{K}'$ is a *counterfactual* for P if $\mathcal{K}'$ is a counterfactual candidate for P in $\mathcal{K}$ and $\delta_{cf}(\mathcal{K}, \mathcal{K}') \leq \delta_{cf}(\mathcal{K}, \mathcal{K}'')$ for every such candidate $\mathcal{K}''$.

Note that the 2nd (closeness of counterfactual to original instance) and the 3rd (sparse feature changes) objectives as specified by Dandl et al. [20] collapse in this criterion of minimal changes, since in $\mathcal{ELH}$, all features are discrete (a concept either holds or doesn't).

Next, we introduce two measures (l_{min} and l_{mean}) of the likeliness (Dandl et al.'s objective 4 [20]) with respect to an individual x among different CFs fulfilling a removal request. Let I_n denote the set of all *existing negative* individuals $y \in N_I$ w.r.t. C, i.e., the set of individuals y such that $\mathcal{K} \not\models C(y)$. The min-likeliness $l_{\min}$ is the minimal edit distance between x and another individual y such that $\mathcal{K}' \not\models C(y)$; the mean-likeliness l_{mean} denotes the *average* edit distance between x and all such individuals y. Let $P = \langle C(x), rem\rangle$, $\mathcal{K}$ be a KB and $\mathcal{K}'$ be such that $\mathcal{K}' \vdash P$. Moreover, let $y \in I_n$ and $\delta_{lm}(x, y, \mathcal{K}') := |\Delta(A^x_{\mathcal{K}'}, A^y_{\mathcal{K}'})|$. Then, $l_{\min}$ and l_{mean} are defined as follows:

$$l_{\min}(P, \mathcal{K}') := \min_{y \in I_n} \delta_{lm}(x, y, \mathcal{K}') \quad (2)$$

$$l_{\text{mean}}(P, \mathcal{K}') := \frac{1}{|I_n|} \sum_{y \in I_n} \delta_{lm}(x, y, \mathcal{K}') \quad (3)$$

4.1 Generation of Counterfactuals

Note that we distinguish between two types of counterfactual requests depending on whether a concept should hold ($U = add$) or should not hold ($U = rem$). In order to fulfill a request $P = \langle C(x), rem\rangle$ for an assertion $C(x)$, we present Algorithm 1. It is worth mentioning that axiom pinpointing for $\mathcal{EL}$ [9–11] computes all justifications for the entailment of a subsumption ($X \sqsubseteq Y$) and not an assertion ($Y(x)$). Whereas, we are not interested in finding every minimal subset of the TBox entailing subsumption but rather in determining whether subsumption holds for specific concepts in the ABox. Algorithm 1 applies a direct approach to compute all updates of an ABox

Algorithm 1. Assume that $C = C_1 \sqcap C_2 \ldots \sqcap C_n$ and no C_j contains an intersection on the outer level anymore (i.e., $C_j \neq A \sqcap B$ for any C_j). Then Algorithm 1 begins by splitting C into the subconcepts C_j (Line 6) in order to apply the subroutine from Algorithm 2 to each C_j. The sets of concepts added to the global variable *candidates* by `find_candidates`$(\mathcal{K}, \{C_j\}, \emptyset, \emptyset, x)$ are the desired concepts for removal to create one candidate from each set. In other words, if $\mathcal{K}'$ is obtained from $\mathcal{K}$ by removing assertions as described in Lines 10–18, then $\mathcal{K}' \not\models C(x)$. This is due to the reason that the subroutine `find_candidates`$(\mathcal{K}, \{C_j\}, \emptyset, \emptyset, x)$ begins by finding all concepts subsumed by C_j (the set $\mathcal{E}$) as well as all those for which the ABox contains an assertion (the set $\mathcal{D}$). In particular, $\mathcal{D}$ includes all the concepts that are essential while inferring $\mathcal{K} \models C_j(x)$. As a result, each atomic and existential restriction in $\mathcal{D}$ is directly added to the set of relevant concepts since removing the ABox assertions corresponding to each of these concepts can fail the inference $\mathcal{K} \models C_j(x)$. In the case of existential restrictions $\exists r.A$, it does not matter if A is an atomic concept, another existential restriction, or an intersection, since to get local counterfactuals we simply remove the assertions $r(x, a)$ where $\mathcal{K} \models A(a)$ and do not need to remove $A(x)$. Moreover, for each $E_j \in \mathcal{E}$ such that $E_j = E_{j_1} \sqcap \ldots \sqcap E_{j_n}$, removing an assertion for only one concept from E_j can result in failing the inference $\mathcal{K} \models C_j(x)$. The choice of allowing only one concept removal from E_j is essential to avoid redundancy since $\mathcal{K} \not\models E_j(x)$ iff $\mathcal{K} \not\models E_{j_i}(x)$ for at least one $E_{j_i} \in E_j$. The algorithm implements this by considering all possible combinations for different j but taking only one concept from each E_j (Line 12). Moreover, the set of visited concepts is updated to avoid rechecking for concepts that are already visited (Lines 13–15). Finally, the recursive calls to `find_candidates`$(\mathcal{K}, P, visited, relevant, x)$ for each combination P add further concepts as being relevant and, once completed, all the relevant concepts are returned as a candidate set. This recursion

```
1  Input: KB $\mathcal{K}$ and a CF request $P = \langle C(x), rem\rangle$ s.t. $\mathcal{K} \nvdash P$
2  Output: Local counterfactuals $\mathcal{K}'$ in $\mathcal{K}$ sorted by likeliness
     such that with $\mathcal{K}' \vdash P$, global variable $candidates$
3  Function create_candidates_neg($\mathcal{K}$, $C$, $x$):
4      $KBs \leftarrow \emptyset$
5      $candidates \leftarrow \emptyset$
6      $C \leftarrow \{C_1, \ldots, C_n\}$
7      for $C_j$ in $C$ do
8          find_candidates($\mathcal{K}$, $\{C_j\}$, $\emptyset$, $\emptyset$, $x$)
9      end
10     for $\mathcal{X}$ in $candidates$ do
11         $\mathcal{K}_{\mathcal{X}} \leftarrow$ copy($\mathcal{K}$)
12         for $X \in \mathcal{X}$ do
13             if $X \equiv \top$ then
14                 $\mathcal{K}_{\mathcal{X}} \leftarrow None$
15             else if $X$ is an atomic class then
16                 Remove $X(x)$ from $\mathcal{K}_{\mathcal{X}}$
17             else if $X = \exists r.A$ then
18                 Remove all $\{r(x, a) \mid \mathcal{K} \models A(a)\}$ from $\mathcal{K}_{\mathcal{X}}$
19         end
20         $KBs \leftarrow KBs \cup \{\mathcal{K}_{\mathcal{X}}\}$
21     end
22     $cfs \leftarrow \arg\min_{\mathcal{K}' \in KBs} \delta_{cf}(\mathcal{K}, \mathcal{K}')$
23     $cfs\_min \leftarrow$ sort $cfs$ by $l_{\min}$
24     $cfs\_mean \leftarrow$ sort $cfs$ by $l_{\text{mean}}$
25 return $cfs\_min$, $cfs\_mean$
```

Algorithm 1: Finds all non-redundant updates $\mathcal{K} \to \mathcal{K}'$ s.t. $\mathcal{K}' \not\models C(x)$

```
1  Input: KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, set of concepts to look at in this
     iteration ($c\_set$), set of concepts already visited, set of
     visited concepts relevant for this candidate, target
     individual $x$, global variable $candidates$
2  Function find_candidates($\mathcal{K}$, $c\_set$, $visited$, $relevant$, $x$):
3      $relevant \leftarrow relevant \cup c\_set$
4      $\mathcal{E} \leftarrow \{E | \exists F : (E \sqsubseteq F) \in \mathcal{T}, \mathcal{K} \models E \sqsubseteq C_i$ for some
         $C_i \in c\_set\}$
5      $\mathcal{D} \leftarrow \{D | D \in \mathcal{E}, D \notin visited$ and $D(x) \in \mathcal{A}\}$
6      for each $D_j \in \mathcal{D}$ do
7          $relevant \leftarrow relevant \cup \{D_j\}$
8      end
9      for each $E_j = E_{j_1} \sqcap E_{j_2} \sqcap \ldots \sqcap E_{j_n} \in \mathcal{E}$ do
10         $EI_j \leftarrow \{E_{j_1}, \ldots, E_{j_n}\}$
11     end
12     $\mathcal{P} \leftarrow \{P \subseteq \cup_j EI_j \mid \forall_j |P \cap EI_j| \geq 1$ and
         $\nexists p \in P \forall_j |P \setminus \{p\} \cap EI_j| \geq 1\}$
        // all non-redundant combinations covering
         each $EI_j$
13     for each $E \in \mathcal{E}$ do
14         $visited \leftarrow visited \cup \{E\}$
15     end
16     if $\mathcal{P} = \emptyset$ then
17         $candidates \leftarrow candidates \cup \{relevant\}$
            // candidates is a set of sets
18     else
19         for each $P \in \mathcal{P}$ do
20             find_candidates($\mathcal{K}$, $P$, $visited$, $relevant$, $x$)
21         end
22     end
```

Algorithm 2: Recursively follow path of subsumptions, add set(s) of relevant concepts to global variable *candidates*

```
1  Input: KB $\mathcal{K}$ and a CF request $P = \langle C(x), add \rangle$ s.t. $\mathcal{K} \nvdash P$
2  Output: Local counterfactual $\mathcal{K}'$ in $\mathcal{K}$ such that $\mathcal{K}' \vdash P$
3  Function create_candidates_pos($\mathcal{K}$, $C$, $x$):
4      $\mathcal{K}' \leftarrow$ copy($\mathcal{K}$)
5      $C \leftarrow \{C_1, \ldots, C_n\}$
6      for $C_j$ in $C$ do
7          if $C_j$ is an existential restriction $\exists r.D$ then
8              add $r(x, y)$ to $\mathcal{K}'$ for a fresh individual $y$
9              $\mathcal{K}' \leftarrow$ create_candidates_pos($\mathcal{K}'$, $D$, $y$)
                // recursive call
10         else
11             add $C_j(x)$ to $\mathcal{K}'$
12     end
13 return $\mathcal{K}'$
```

Algorithm 3: Updated KB $\mathcal{K}'$ such that $\mathcal{K}' \models C(x)$

for each P is essential since P also includes (removal) candidates for C_j. Finally, Algorithm 1 uses the global variable *candidates* and generates local counterfactuals corresponding to each set $\mathcal{X}$ of candidates. Once all the updated KBs have been enumerated, the algorithm sorts the KBs with the least edit distance as counterfactuals according to Eq 1. Furthermore, among these counterfactuals, the ones with the highest likeliness are selected for creating natural language CEs following the two likeliness measurements (Eqs 2-3).

Theorem 4.5. *The Algorithm 1 is sound and complete: Given a KB $\mathcal{K}$ and an update request $P = \langle C(x), rem \rangle$, such that $\mathcal{K} \nvdash P$, Algorithm 1 returns a collection KBs such that $\mathcal{K}' \in KBs$ if and only if $\mathcal{K}'$ is a local counterfactual for P without redundancy in $\mathcal{K}$.*

Proof. To establish the correctness we first prove that $\mathcal{K}' \vdash P$ is true for every KB $\mathcal{K}'$ returned by Algorithm 1. Since $\mathcal{K}' \in KBs$, there is some candidate $\mathcal{X}$ (Line 10) such that $\mathcal{K}' = \mathcal{K}_{\mathcal{X}}$, that is, $\mathcal{K}'$ is obtained from $\mathcal{K}$ after applying changes due to some candidate set $\mathcal{X}$. Moreover, each candidate $\mathcal{X}$ is returned after the call find_candidates($\mathcal{K}, \{C_j\}, \emptyset, \emptyset, x$) is made for some concept $C_j \in C$. This implies that for each $\mathcal{K}'$, there is some $C_j \in C$ such that $\mathcal{K}' \not\models C_j(x)$ since all the assertions necessary to infer $\mathcal{K} \models C_j(x)$ have been removed (Lines 10–19). Consequently, $\mathcal{K}' \not\models C(x)$. Notice that, Line 17 in Algorithm 1 only removes assertions $r(x, a)$ for a role r rather than also removing $s(x, a)$ for subroles $s \sqsubseteq r$. This is unproblematic since Algorithm 2 (Line 4) already adds concepts of the form $\exists s.A$ in the set of candidates (since $\exists s.A \sqsubseteq \exists r.A$). Consequently, such assertions are removed when Line 17 is triggered in Algorithm 1 for $X = \exists s.A$. The fact that $\mathcal{K}'$ is non-redundant follows from Line 12 in Algorithm 2, since exactly one candidate concept is chosen for conjunctive $E \in \mathcal{E}$. This proves that the KBs returned by Algorithm 1 are indeed local counterfactuals, and each of them fulfills the request P. Finally, Lines 22–24 sort the output KBs according to their edit distance (Eq. 1) and likeliness (Eq. 2-3).

Conversely, let $\mathcal{L}$ be a non-redundant local counterfactual of $\mathcal{K}$. Then $\mathcal{L} \not\models C(x)$ and consequently, $\mathcal{L} \not\models C_j(x)$ for at least one $C_j \in C$. We prove that $\mathcal{L} \in KBs$. Let $R^j_{rem} = \{R_1, \ldots, R_m\}$ denotes the set of all candidates returned by the call find_candidates($\mathcal{K}, \{C_j\}, \emptyset, \emptyset, x$). Clearly, R^j_{rem} includes all possible ways of removing assertions from $\mathcal{K}$ to fail the inference $\mathcal{K} \models C_j(x)$. Then, there must be some $R_i \in R^j_{rem}$ such that $\mathcal{L}$ is obtained from $\mathcal{K}$ by removing assertions corresponding to the candidate R_i since $\mathcal{L} \models C_j$ is true otherwise. We claim that $\mathcal{L}$ is returned in the iteration i when R_i is selected as a candidate in Line 10. Suppose to the contrary that $\mathcal{L}$ is obtained from $\mathcal{K}$ by removing not only the concept assertions corresponding to concepts in R_i but requires some further changes. Without loss of generality, we assume that these changes only include the removal of ABox assertions since adding assertions is redundant. But this implies that either the update $\mathcal{K} \rightarrow \mathcal{L}$ is redundant (if these additional removal concerns concepts not in R_i) or $\mathcal{L}$ is not a local counterfactual candidate for P (if these concern some individual $y \neq x$). This is due to the reason that the removal of assertions for x corresponding to concepts in R_1 suffices to create a KB that fulfills P. This leads to a contradiction since $\mathcal{L}$ is a non-redundant local counterfactual of $\mathcal{K}$ for P. This completes the correctness in this direction and establishes the proof □

Updating $\mathcal{K}$ to fulfill a request $\langle C(x), add \rangle$ is relatively easier to implement, since it suffices to add axioms implying $\mathcal{K} \models C(x)$. This is similar to abduction since for a KB $\mathcal{K}$, a hypotheses set H of assertions such that $\mathcal{K} \cup H \models C(x)$, yields a CE for $C(x)$ in $\mathcal{K}$.

However, we differ from abduction in that we also allow to add fresh individuals. Algorithm 3 presents our approach .

Algorithm 3. The algorithm simply adds assertions $C_j(x)$ to the KB for each $C_j \in C$. This yields the only non-redundant way to infer $\mathcal{K}' \models C(x)$. Notice that if $C_j = \exists r.Z$ for some j and a concept Z, then there are multiple options regarding the object y such that the assertion $r(x, y)$ has to be added. Either a new individual y is added to $\mathcal{K}'$ and the process is repeated for y regarding Z, or y is an existing individual in $\mathcal{K}$ which is already in Z. If there exists no y in $\mathcal{K}$ with $\mathcal{K} \models Z(y)$, of course it is not possible to create a local counterfactual. In our case, we allow the addition of a fresh individual y. This achieves the desired goal since our main focus is on finding CEs for $C(x)$. In practice, the decision to allow fresh individuals may depend on the application context. Moreover, adding an assertion $D(x)$ to $\mathcal{K}'$ is redundant if D is a subconcept of C and a necessary consequence if D is a superconcept of C.

Theorem 4.6. *The Algorithm 3 is sound and complete: Given a KB $\mathcal{K}$ and a request $P = \langle C(x), add\rangle$ such that $\mathcal{K} \nvDash P$, Algorithm Algorithm 3 returns an updated KB $\mathcal{K}'$ such that $\mathcal{K}'$ is a counterfactual for P without redundancy in $\mathcal{K}$.*

Proof. We first prove that $\mathcal{K}' \vdash P$ is true for $\mathcal{K}'$ returned by Algorithm 3. As before, assume without loss of generality that C can be written as a conjunct $C_1 \sqcap \ldots \sqcap C_n$ and no C_j contains an intersection on the outer level anymore. Clearly, $\mathcal{K}' \models C_j(x)$ is true once Lines 7–11 have been executed for each $C_j \in C$. This implies that $\mathcal{K}' \models C(x)$. Moreover, the applied changes are non-redundant, minimal, and only local. Conversely, suppose there exists $\mathcal{L} = (\mathcal{T}, \mathcal{A}'')$ such that $\mathcal{L} \models C(x)$. Suppose that $\mathcal{L} \neq \mathcal{K}'$. Clearly, Lines 6–11 state the necessary conditions to infer that $\mathcal{L} \models C_j(x)$ for a KB $\mathcal{L}$. This implies that $\mathcal{L}$ applies not only these changes but adds or removes further assertions to the ABox $\mathcal{A}''$. Then a similar line of reasoning as in the proof of Theorem 4.5 implies that either the update $\mathcal{K} \longrightarrow \mathcal{L}$ is redundant or $\mathcal{L}$ is not a local counterfactual of $\mathcal{K}$ for P. □

Notice that, an obvious question emerges when considering inferred knowledge: how to treat the implicit assertions in an Abox for the selection of CEs? In other words, should the implicit knowledge also be considered when measuring the edit distance? This leads one to chose that $\mathcal{A}$ should be fully materialized. One rationale behind materialization is: 1) all "features" (as authors in [20] called it) that have to be removed or added to generate the counterfactual, even if implicit, should be counted in finding the counterfactual with least "feature changes". 2) Implicit features that do not have to be removed to create the counterfactual should not be removed. For example, assume $\mathcal{T} \models A \sqsubseteq B \sqsubseteq C \sqsubseteq D$, $\mathcal{A} \models A(x)$, $P = \langle C(x), rem\rangle$, then $A(x), B(x), C(x)$ have to be removed and counted to the edit distance, but $D(x)$ does not. Materializing the KB before applying our algorithm achieves that result. Nevertheless, this decision is optional and may depend on specific applications rather than being obligatory when utilizing our algorithm.

Table 2: Overview of the final, modified datasets in terms of number of instances (N_I), axioms, atomic concepts and roles.

Dataset	Instances	Axioms	Atomic Concepts	Roles
Family	202	2,033	18	5
Animals	28	170	19	4

5 EXPLANATIONS PREFERRED BY USERS

We conducted a survey in which we let participants rate different potential CEs against each other. We then compared the participants' preferences with the decisions made by our approach. Our hypothesis was that CEs generated by our algorithm will get positive ratings by study participants. We used modified versions of the *Family* and *Animals* ontologies [55] to evaluate our approach and materialized the ABoxes. These ontologies were chosen because the concept, role and individual names therein are familiar and understandable to average lay users—in contrast to, for example, ontologies related to bio-medicine or chemistry. We used the DL concept learner [16] with ELTL—the $\mathcal{EL}$ Tree Learner [15]—to learn the concepts to be used for counterfactual generation, since a future goal is to combine these programs to reach a fully automated explainable AI.

5.1 Data Generation

Table 2 gives an overview of the datasets used for our user survey. We used the DL concept learner [16] with ELTL to learn concepts from the Family and Animals ontologies that describe classes of family members or species of animals, respectively. A detailed description of the datasets and how we applied ELTL to find concepts for explaining can be found in the repository [1]. For the study, we created a CE from each counterfactual candidate that could be drawn from the learned concepts, the user ratings of the CEs selected by our algorithm with user ratings of other possible explanations. To keep it consistent for the participants, we also presented explanations for the Family concepts for `Brother` and `Grandmother` as the corresponding concepts to `Sister` and `Grandfather` in the survey, even if ELTL did not correctly recognize these concepts and therefore our algorithm was not applied here.

5.2 Setup of User Survey

Using the generated concepts from the Family and Animals ontologies (see above) and their respective counterfactual candidates, we generated short stories of artificial intelligences classifying people in a family tree or animals and created a CE from each counterfactual candidate. We conducted an online survey via SoSciSurvey in German. Participants were recruited through social networks and snowballing. At first, participants were informed about the content and goal of the survey and what CEs are, and later, the CEs were presented. First, a scenario was described in which an AI would classify instances of family members or animals. Then, on every page, a classification made by an AI was presented in one sentence, followed by one or multiple sentences giving CEs for the classification, e.g. "I would not have classified this animal as a turtle, if it did not have scales". Within the two scenarios, classifications

Table 3: Overview of alignment of algorithm decisions with user ratings as ground truth. Notation: T = True, F = False, P = Positive, N = Negative

$l_{\min}$	TP	TN	FP	FN	F1-Score
Family	4	4	0	0	1.0
Animals	3	6	5	7	0.33
l_{mean}	TP	TN	FP	FN	F1-Score
Family	4	4	0	0	1.0
Animals	3	8	3	7	0.37

were presented in randomized order, one on each page. For the Family ontology, where each concept had led to two counterfactual candidates, the participants were randomly shown only one of the CEs. Because many explanations were quite similar (e.g. all concepts included counterfactual candidates referring to gender) it was made sure that they were presented mixed explanation types. Each explanation was accompanied by one item asking to rate on a scale from one to seven how helpful they perceived the explanation for understanding the decision of the program. For the animals scenario, participants were shown all counterfactual candidates (between two and five) at the same time, in random order, and presented the same rating scale for each of the explanations.

5.3 Results of User Survey

In the following, we present the results of our evaluation of our CE algorithm through a user survey.

Sample. 72 people took part in the survey. Age ranged between 20 and 69 (mean = 34.9, median = 32, standard deviation = 12.1, missing age data for one participant). 30 participants were female, 39 male and 3 diverse. Participants had mixed professions including both academic and non-academic ones, technical and non-technical.

5.3.1 *User ratings.*

We used Wilcoxon signed-rank tests to calculate significance of deviation from the central value (4 on a scale from 1-7) of the rating of helpfulness for understanding for each CE. Tests were chosen given the fact that we use ad-hoc generated items, so we assume the ratings to be ordinal. For all six concepts, users preferred the explanation that featured a role referring to relatives of the individual (`hasChild`, `hasSibling`) against the explanation that featured a C_i referring to the individuals `gender` (all p<.001, except `Sister`: p<.01). These explanations (and only these) differed significantly from the central value (all p<.001). For the Animaly ontology, we used Wilcoxon signed-rank tests as above (but for matched samples). For concepts with more than two counterfactuals, we used the Friedmann test. If present, participants always rated the CE mentioning an animal laying eggs as helpful (p<.01). Apart from that, decisions on this ontology showed no clear pattern, choosing (with p<.01) different features for different animals for explanation.

5.3.2 *Comparison of algorithmic decisions with user ratings.*

The match of our algorithms' decision with the participants' ratings can be seen in tablet 3. We counted all CEs where participant ratings were significantly positive, so both algorithm and participants sometimes selected more than one explanation per scenario. Details on explanations selected by algorithm and participants can be found in our repository [1].

5.3.3 *Interpretation of results.*

This study compared different CEs, i.e. explanations of the type "what features of x needed to be different for x not to be a C?" from user perspective. Overall, we suspect that participants preferred explanations referring to features that are rather unlikely and therefore more characteristic of the person or animal, though more studies would be needed here. In contrast, features which are very common (like being of a certain gender or having legs) were chosen less. This fits our likeliness measurement idea, since removing a feature that does not appear often in the population for candidate generation should also result in a rather high likeliness score using our definitions. Our algorithm partly manages to cover that, but could be improved.

6 DISCUSSION

We showed that our approach can generate counterfactuals with minimal *edit distance* measured by axiom additions and removals i.e. few features changes to the individual. Moreover, as there can be multiple counterfactuals per individual with minimal edit distance, we explored two *likeliness* measures to choose among them.

Regarding the choice of the best counterfactual for an explanation, the results of our evaluation survey show room for improvement. In future work, a "learning to rank" algorithm might be used to automatically learn a likeliness measure. Moreover, $\mathcal{ELH}$ is a rather restricted DL. We plan to expand our algorithm to the more complex DL $\mathcal{ALC}$. The main challenges for $\mathcal{ALC}$ include negation and disjunctions. In $\mathcal{ALC}$, counterfactuals for $\langle C(x), add\rangle$ would differ further from abduction, since the ABox abduction does not return any explanation if $\mathcal{K} \models \neg C(x)$ but there might exist CFs obtained by removing the entailment of $\neg C(x)$ from $\mathcal{K}$.

Structure of ontologies. A point open for discussion addresses how to deal with different structures of ontologies. As it is the case in the Family ontology, sometimes individuals might or might not have the same roles, while it is very unlikely for them to have the same objects (e.g. children) for these roles, which is affecting the likeliness measurement. Furthermore, it might depend on the ontology if sub-features of the changed feature should be counted into the edit distance, as we did. Thus, the usefulness of different distance calculation possibilities may depend on use cases and ontology structures.

Actionability. While we tried to check for plausibility of counterfactual instances, our scoring mechanism cannot make sure yet that the axioms that were changed can actually be changed in the real world. One argument for CEs is that in many applications it might be interesting for data subjects to get to know how they can change their classification [54]. However, the Family ontology shows an example of cases where this is not possible, since people cannot usually change their gender or relatives. Poyiadzi et al. [45] discuss the relevance of actionability of counterfactuals. Our future work will put more focus on applications where actionability can be reached and how to do this.

Applications. The Family ontology contains data about people's features and relations, as in DBpedia, YAGO, and Wikidata. CEs might be especially useful when humans are the classification target and demand a transparent justification of the result. To enrich KBs, additional information can be extracted from the web. Concept learning allows checking the consistency of the extracted information and inferring new (implicit) axioms from the explicitly stated axioms. Concept learning has been effectively applied to medical ontologies [40], but the learned concepts can become very long [38], making them hard to grasp even for experts. Counterfactuals, which might even be verbalized in natural language, help to steer focus to the most important parts of the concept. Ultimately, we want to develop a chatbot that, in the spirit of XAI, provides users with natural language explanations of automatically learned concepts and can be applied to various use cases in areas including web science, medicine and finance.

7 CONCLUSION

We propose the first approach for generating CEs for concept assertions in the DL $\mathcal{ELH}$. Our approach performs well on the objective to generate counterfactual candidates which are similar to the individual. We discussed possibilities to improve the likeliness measurement of counterfactuals in accordance with findings from a user study. Our future work will move on to more complex DLs.

REFERENCES

[1] Amina Adadi and Mohammed Berrada. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160.

[2] Alejandro Barredo Arrieta, Natalia Díaz Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* 58 (2020), 82–115.

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *ISWC/ASWC (LNCS, Vol. 4825)*. Springer, 722–735.

[4] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider (Eds.). 2003. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press.

[5] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. 2017. *An Introduction to Description Logic.* Cambridge University Press.

[6] Franz Baader, Patrick Koopmann, Francesco Kriegel, and Adrian Nuradiansyah. 2021. Computing Optimal Repairs of Quantified ABoxes w.r.t. Static $\mathcal{EL}$ TBoxes. In *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings.* 309–326. https://doi.org/10.1007/978-3-030-79876-5_18

[7] Franz Baader, Patrick Koopmann, Francesco Kriegel, and Adrian Nuradiansyah. 2022. Optimal ABox Repair w.r.t. Static *EL* TBoxes: From Quantified ABoxes Back to ABoxes. In *The Semantic Web - 19th International Conference, ESWC 2022.* 130–146.

[8] Franz Baader, Francesco Kriegel, Adrian Nuradiansyah, and Rafael Peñaloza. 2018. Making Repairs in Description Logics More Gentle. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018.* 319–328. https://aaai.org/ocs/index.php/KR/KR18/paper/view/18056

[9] Franz Baader and Rafael Peñaloza. 2010. Axiom Pinpointing in General Tableaux. *J. Log. Comput.* 20, 1 (2010), 5–34. https://doi.org/10.1093/logcom/exn058

[10] Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn. 2007. Pinpointing in the Description Logic EL. In *Proceedings of the International Workshop on Description Logics (DL 2007) (CEUR Workshop Proceedings, Vol. 250)*, Diego Calvanese, Enrico Franconi, Volker Haarslev, Domenico Lembo, Boris Motik, Anni-Yasmin Turhan, and Sergio Tessaris (Eds.). CEUR-WS.org.

[11] Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn. 2007. Pinpointing in the Description Logic EL^+. In *KI 2007: Advances in Artificial Intelligence, 30th Annual German Conference on AI, KI 2007, Proceedings (Lecture Notes in Computer Science, Vol. 4667)*, Joachim Hertzberg, Michael Beetz, and Roman Englert (Eds.). Springer, 52–67.

[12] Meghyn Bienvenu and Camille Bourgaux. 2016. Inconsistency-Tolerant Querying of Description Logic Knowledge Bases. In *Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering - 12th International Summer School 2016, Aberdeen, UK, September 5-9, 2016, Tutorial Lectures (Lecture Notes in Computer Science, Vol. 9885)*, Jeff Z. Pan, Diego Calvanese, Thomas Eiter, Ian Horrocks, Michael Kifer, Fangzhen Lin, and Yuting Zhao (Eds.). Springer, 156–202. https://doi.org/10.1007/978-3-319-49493-7_5

[13] Roi Blanco, Giuseppe Ottaviano, and Edgar Meij. 2015. Fast and Space-Efficient Entity Linking for Queries. In *WSDM.* ACM, 179–188.

[14] Sebastian Brandt. 2004. Reasoning in ELH wrt general concept inclusion axioms. *Institute for Theoretical Computer Science, Dresden University of Technology* (2004).

[15] Lorenz Bühmann, Daniel Fleischhacker, Jens Lehmann, André Melo, and Johanna Völker. 2014. Inductive Lexical Learning of Class Expressions. In *EKAW (LNCS, Vol. 8876).* Springer, 42–53.

[16] Lorenz Bühmann, Jens Lehmann, and Patrick Westphal. 2016. DL-Learner - A framework for inductive learning on the Semantic Web. *J. Web Semant.* 39 (2016), 15–24.

[17] Xuqing Chai. 2020. Diagnosis Method of Thyroid Disease Combining Knowledge Graph and Deep Learning. *IEEE Access* 8 (2020), 149787–149795.

[18] Jiaoyan Chen, Xi Chen, Ian Horrocks, Erik B. Myklebust, and Ernesto Jiménez-Ruiz. 2020. Correcting Knowledge Base Assertions. In *WWW.* ACM / IW3C2, 1537–1547.

[19] Yu-Chi Chung, RJ Kuo, et al. 2023. A domain adaptation approach for resume classification using graph attention networks and natural language processing. *Knowledge-Based Systems* 266 (2023), 110364.

[20] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. 2020. Multi-Objective Counterfactual Explanations. In *PPSN (1) (LNCS, Vol. 12269).* Springer, 448–469.

[21] Warren Del-Pinto and Renate A. Schmidt. 2019. ABox Abduction via Forgetting in ALC. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019.* 2768–2775.

[22] Caglar Demir and Axel-Cyrille Ngonga Ngomo. 2021. DRILL- Deep Reinforcement Learning for Refinement Operators in ALC. *CoRR* abs/2106.15373 (2021).

[23] Jianfeng Du, Hai Wan, and Huaguan Ma. 2017. Practical TBox Abduction Based on Justification Patterns. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.* 1100–1106. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14402

[24] Giorgos Filandrianos, Konstantinos Thomas, Edmund Dervakos, and Giorgos Stamou. 2022. Conceptual Edits as Counterfactual Explanations. In *Proceedings of the AAAI 2022 Spring Symposium on Machine Learning and Knowledge Engineering for Hybrid Intelligence (AAAI-MAKE 2022) (CEUR Workshop Proceedings, Vol. 3121)*, Andreas Martin, Knut Hinkelmann, Hans-Georg Fill, Aurona Gerber, Doug Lenat, Reinhard Stolle, and Frank van Harmelen (Eds.). CEUR-WS.org. https://ceur-ws.org/Vol-3121/paper6.pdf

[25] Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M. Suchanek. 2017. Predicting Completeness in Knowledge Bases. In *WSDM.* ACM, 375–383.

[26] Aldo Gangemi, Andrea Giovanni Nuzzolese, Valentina Presutti, Francesco Draicchio, Alberto Musetti, and Paolo Ciancarini. 2012. Automatic Typing of DBpedia Entities. In *ISWC (1) (LNCS, Vol. 7649).* Springer, 65–81.

[27] Brigitte Grau and Anne-Laure Ligozat. 2018. A Corpus for Hybrid Question Answering Systems. In *WWW (Companion Volume).* ACM, 1081–1086.

[28] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2019. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* 51, 5 (2019), 93:1–93:42.

[29] Ken Halland, Arina Britz, and Szymon Klarman. 2014. TBox Abduction in ALC Using a DL Tableau. In *Informal Proceedings of the 27th International Workshop on Description Logics, Vienna, Austria, July 17-20, 2014.* 556–566. https://ceur-ws.org/Vol-1193/paper_42.pdf

[30] Stefan Heindorf, Lukas Blübaum, Nick Düsterhus, Till Werner, Varun Nandkumar Golani, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. 2022. EvoLearner: Learning Description Logics with Evolutionary Algorithms. In *WWW.* ACM, 818–828.

[31] Matthew Horridge. 2011. *Justification based explanation in ontologies.* Ph. D. Dissertation. University of Manchester, UK. http://www.manchester.ac.uk/escholar/uk-ac-man-scw:131699

[32] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *NeurIPS.*

[33] Luigi Iannone, Ignazio Palmisano, and Nicola Fanizzi. 2007. An algorithm based on counterfactuals for concept learning in the Semantic Web. *Appl. Intell.* 26, 2 (2007), 139–159. https://doi.org/10.1007/s10489-006-0011-5

[34] Aditya Kalyanpur, Bijan Parsia, and Bernardo Cuenca Grau. 2006. Beyond Asserted Axioms: Fine-Grain Justifications for OWL-DL Entailments. In *Proceedings of the 2006 International Workshop on Description Logics (DL 2006), Windermere, Lake District, UK, May 30 - June 1, 2006 (CEUR Workshop Proceedings, Vol. 189)*, Bijan Parsia, Ulrike Sattler, and David Toman (Eds.). CEUR-WS.org.

[35] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. 2007. Finding All Justifications of OWL DL Entailments. In *ISWC/ASWC (Lecture Notes in*

Computer Science, Vol. 4825). Springer, 267–280.

[36] Patrick Koopmann. 2021. Signature-Based Abduction with Fresh Individuals and Complex Concepts for Description Logics. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*. 1929–1935.

[37] N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. 2021. Neural Class Expression Synthesis. *CoRR* abs/2111.08486 (2021).

[38] N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. 2022. Learning Concept Lengths Accelerates Concept Learning in ALC. In *ESWC (LNCS, Vol. 13261)*. Springer, 236–252.

[39] Jonathan Lajus and Fabian M. Suchanek. 2018. Are All People Married?: Determining Obligatory Attributes in Knowledge Bases. In *WWW*. ACM, 1115–1124.

[40] Jens Lehmann and Pascal Hitzler. 2010. Concept learning in description logics using refinement operators. *Mach. Learn.* 78, 1-2 (2010), 203–250.

[41] Karsten Lehmann and Anni-Yasmin Turhan. 2012. A Framework for Semantic-Based Similarity Measures for $\mathcal{ELH}$ -Concepts. In *JELIA (LNCS, Vol. 7519)*. Springer, 307–319.

[42] Christian Meske, Enrico Bunde, Johannes Schneider, and Martin Gersch. 2022. Explainable artificial intelligence: objectives, stakeholders, and future research opportunities. *Information Systems Management* 39, 1 (2022), 53–63.

[43] Christoph Molnar. 2020. *Interpretable machine learning*. Lulu. com. christophm.github.io/interpretable-ml-book/

[44] Carina Negreanu, Alperen Karaoglu, Jack Williams, Shuang Chen, Daniel Fabian, Andrew D. Gordon, and Chin-Yew Lin. 2022. Rows from Many Sources: Enriching row completions from Wikidata with a pre-trained Language Model. In *WWW (Companion Volume)*. ACM, 1272–1280.

[45] Rafael Poyiadzi, Kacper Sokol, Raúl Santos-Rodríguez, Tijl De Bie, and Peter A. Flach. 2020. FACE: Feasible and Actionable Counterfactual Explanations. In *AIES*. ACM, 344–350.

[46] Júlia Pukancová and Martin Homola. 2018. ABox Abduction for Description Logics: The Case of Multiple Observations. In *Proceedings of the 31st International Workshop on Description Logics 2018), Tempe, Arizona, US, October 27th - to - 29th, 2018*. https://ceur-ws.org/Vol-2211/paper-31.pdf

[47] Charlotte Rudnik, Thibault Ehrhart, Olivier Ferret, Denis Teyssou, Raphaël Troncy, and Xavier Tannier. 2019. Searching News Articles Using an Event Knowledge Graph Leveraged by Wikidata. In *WWW (Companion Volume)*. ACM, 1232–1239.

[48] Md. Kamruzzaman Sarker and Pascal Hitzler. 2019. Efficient Concept Induction for Description Logics. In *AAAI*. AAAI Press, 3036–3043.

[49] Stefan Schlobach and Ronald Cornet. 2003. Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. In *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence 2003*. Morgan Kaufmann, 355–362. http://ijcai.org/Proceedings/03/Papers/053.pdf

[50] Ilia Stepin, José Maria Alonso, Alejandro Catalá, and Martin Pereira-Fariña. 2021. A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence. *IEEE Access* 9 (2021), 11974–12001.

[51] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*. ACM, 697–706.

[52] Damian Szklarczyk, Annika L. Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T. Doncheva, John H. Morris, Peer Bork, Lars Juhl Jensen, and Christian von Mering. 2019. STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res.* 47, Database-Issue (2019), D607–D613.

[53] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.

[54] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2018. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.* 31 (2018), 841.

[55] Patrick Westphal, Lorenz Bühmann, Simon Bin, Hajira Jabeen, and Jens Lehmann. 2019. SML-Bench - A benchmarking framework for structured machine learning. *Semantic Web* 10, 2 (2019), 231–245.

[56] David S. Wishart, Yannick D. Feunang, An Chi Guo, Elvis J. Lo, Ana Marcu, Jason R. Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, Nazanin Assempour, Ithayavani Iynkkaran, Yifeng Liu, Adam Maciejewski, Nicola Gale, Alex Wilson, Lucy Chin, Ryan Cummings, Diana Le, Allison Pon, Craig Knox, and Michael Wilson. 2018. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.* 46, Database-Issue (2018), D1074–D1082.

[57] Hamada M. Zahera, Stefan Heindorf, and Axel-Cyrille Ngonga Ngomo. 2021. ASSET: A Semi-supervised Approach for Entity Typing in Knowledge Graphs. In *K-CAP*. ACM, 261–264.

[58] Qing Zhan and Hang Yin. 2018. A loan application fraud detection method based on knowledge graph and neural network. In *Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence*. 111–115.