Explaining Concepts in Description Logic through Counterfactual Reasoning

Leonie Nora Sieger[1 [0000−0002−5137−725X]], Stefan Heindorf[1 [0000−0002−4525−6865]], Lukas Blübaum, Yasir Mahmood[1 [0000−0002−5651−5391]], and Axel-Cyrille Ngonga Ngomo[1 [0000−0001−7112−3516]]

Paderborn University, Germany

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 in description logics. 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. In a second study, we explore possible use cases for counterfactual explanations.

Keywords: XAI, machine learning, description logic, ontologies

1 Introduction

Knowledge bases (KBs) are commonly used to represent information in various domains. KBs such as DBpedia [3], Wikidata [43], or YAGO [40] are used in web applications, including information retrieval [36], information generation [33], web search [8] and question answering [21]. In the medical domain, KBs such as DrugBank [46], Snomed [9], and STRING [41] are widely used for predicting whether a molecule is safe, whether it helps against a certain disease, and what the functions of proteins are [24]. Further applications include

medical diagnostics [12], credit scoring [48] and hiring decisions [42]. Typical machine learning tasks include predicting whether the information about an entity is complete [19,29], correct [13], and what category an entity falls in [47,20]. In many of these domains, explaining algorithmic decisions of AI systems to stakeholders is important [1,22,2,31]: (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 [10,16,27,28,23]. 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 [23]. 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 [32]. CEs focus on an antecedent that would have caused a different outcome (classification) had it been the case [39].

Counterfactual explanations answer the question of how the classifier's input needs to be minimally changed to arrive at a different prediction [44]. Dandl et al. [14] generalize this idea and take further criteria into account. They propose a multi-objective optimization problem with four objectives: (1) the prediction for the CE should be as close as possible to the desired prediction; (2) the CE should be as similar as possible to the original instance; (3) feature changes should be sparse; (4) the CE should have plausible/likely feature values/combinations.

While CEs are widely used for tabular data [14,39,44], in this paper we transfer the notion of CEs to DLs and generate simple explanations for *individuals*. Our main contribution is a definition of CEs in the general setting of the description logic knowledge bases. Given a 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 [44,14], 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 conducted a user survey to investigate if the selected CEs are indeed preferred by users. Finally, we also conducted a study comparing CEs and simple

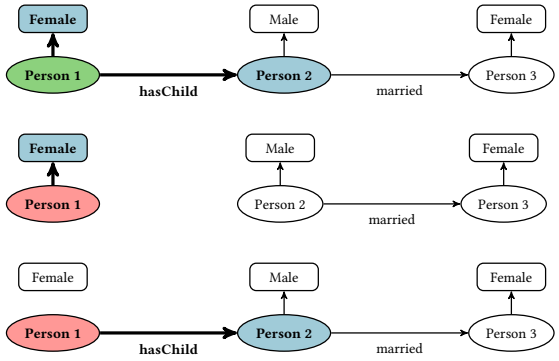

Fig. 1. The concept $\texttt{Female} \sqcap \exists \texttt{hasChild}.\top$ classifies Person 1 as a mother in KB 1 (green), while its corresponding counterfactuals in KBs 2 and 3 (red) are not classified as such.

concept-based explanations in different scenarios to explore possible uses of CE. 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 DL.
3. We provide a heuristic (likeliness) to choose among multiple counterfactuals such that the user will only see the most useful counterfactual explanations.
4. We evaluate our CEs and likeliness measures via a user study.
5. We investigate practical application scenarios for counterfactual explanations

In what follows, Section 2 discusses related work, Section 3 introduces preliminaries, Section 4 formalizes the notion of CEs in DL and presents our algorithms to generate CEs, and Section 5 presents the user study to investigate user preferences. Section 6 describes the study comparing concept based and counterfactual explanations. Finally, Section 7 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

Dervakos et al. [17] provided an algorithm for counterfactual explanations using DL and knowledge graphs. In their work, they aim to find counterfactuals for any classifier by using a knowledge graph of a specific structure which corresponds to the data used by the classifier. In this knowledge graph, they find all edits between the individual they want to find a counterfactual to and each individual where the classifier comes to the desired prediction. Thus, they do not find counterfactuals as defined by Wachter [44] or Dandl [14], since they might not

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

be minimal. Their edit distance approach is similar to our likeliness (see definition 4). However, in our approach we find counterfactuals with minimal edits and our likeliness measurement is just an additional tool to find the most useful explanation in the case that there are multiple minimal counterfactuals. Further, the authors demand a very simple knowledge base, while our approach provides counterfactuals for complex DL concepts. To summarize, Dervakos et al. [17] target a wider range of application scenarios, while we specify on explaining one type of classifier, having way less data requirements, allowing a more expressive DL and finding 100% accurate and minimal counterfactuals, from which we select user-friendly explanations. Additionaly, given their data requirements, our likeliness measurements could also be used analogous for their application context.

Counterfactuals have been mentioned in the context of description logics before that, but their definitions are vastly different from ours. Filandrianos et al. [18] 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 [10,37,16,27,28,23]. Iannone and colleagues [25] use the term "counterfactual" for negated residuals (i.e., parts of a *concept*) and use them to prune the search space of a 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 which parts of description logics we support. For further details, we refer the reader to existing literature [9,5].

Table 1. Used description logic constructs.

	Syntax	**Semantics**	**Construct**
Concepts	$\top$	$\Delta^{\mathcal{I}}$	top concept
	C, D	$C^{\mathcal{I}}, D^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$	concepts
	r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	roles
	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}$ 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

Description Logic 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. Existential restrictions ($\exists r.C$) can be built from the concept descriptions C, D and roles r, s.

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 $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ [30]. Furthermore, each individual $x \in N_I$, where N_I is a finite set of individual names, is assigned an element $x^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ [30].

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 $x \neq y$ [5,4].

The concepts that counterfactuals are generated for do not have to be part of the KB and also can be constructed using intersections ($C \sqcap D$). However, the KB is restricted to contain only non-complex concepts, i.e. without intersections or unions. See section 7 for reasons for this choice.

4 Counterfactuals in Description Logic

Following Wachter et al. [44] and Dandl et al. [14], 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 creation of a counterfactual candidate can also be denoted

as an update $\mathcal{K} \to \mathcal{K}'$. To update a KB (when $U = rem$), it is often necessary to remove multiple axioms from which the target assertion can be inferred.

Example 1. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T} := \{C \sqsubseteq D\}$ and $\mathcal{A} := \{B(x), C(x), D(x)\}$. Moreover let $P = \langle D(x), rem \rangle$ be a counterfactual request. The updates, $\mathcal{K}'_1 := (\mathcal{T}, \{B(x)\})$ and $\mathcal{K}'_2 := (\mathcal{T}, \{\emptyset\})$ both fulfill P.

Clearly, candidates *without redundancy* having *minimal changes* are preferred in scenarios demanding explanations. Intuitively, an update $\mathcal{K} \to \mathcal{K}'$ 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 $\mathcal{A}'$). We write $\Delta(\mathcal{A}, \mathcal{A}')$ to denote the symmetric difference of $\mathcal{A}$ and $\mathcal{A}'$ defined as $\Delta(\mathcal{A}, \mathcal{A}') := (\mathcal{A} \setminus \mathcal{A}') \cup (\mathcal{A}' \setminus \mathcal{A})$.

Definition 1 (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 $\mathcal{K} \to \mathcal{K}'$ is non-redundant.*

In Example 1, the update $\mathcal{K} \to \mathcal{K}'_1$ is non-redundant, whereas the update $\mathcal{K} \to \mathcal{K}'_2$ is redundant.

When creating a CF 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, CF 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 2 (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. [14], 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.

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

Input: KB $\mathcal{K}$, Concept C (no conjunctions), Individual x such that $\mathcal{K} \models C(x)$
Output: KB $\mathcal{K}$ such that $\mathcal{K} \not\models C(x)$
Function `create_candidates_neg(`$\mathcal{K}$, *c_set*, x`):`
  if $C \equiv \top$ **then**
    $\mathcal{K} \leftarrow None$
  else if C *is an atomic class* **then**
    Remove all $\{D(x) \mid \mathcal{K} \models D(x)$ and $\mathcal{K} \models D \sqsubseteq C\}$ from $\mathcal{K}$
  else if $C = \exists r.A$ **then**
    Remove all $\{r'(x, a) \mid \mathcal{K} \models A(a)$ and $\mathcal{K} \models r' \sqsubseteq r\}$ from $\mathcal{K}$
return ($\mathcal{K}$)

Algorithm 1: Updates KB $\mathcal{K}$ such that $\mathcal{K} \not\models C(x)$

Definition 3 (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. [14] collapse in this criterion of minimal changes, since in DLs, all features are discrete (a concept either holds or doesn't - except for specific DLs which include an operator that introduces uncertainity). Next, we introduce two measures (l_{min} and l_{mean}) of the likeliness (Dandl et al.'s 4th objective [14]) 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:

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

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

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 2 that creates a counterfactual candidate for every concept C_i intersected in the input concept C (and only one candidate in the case that there are no intersections in C). Then, Algorithm 2 breaks the entailment of each $C_i(x)$ to obtain a candidate CF. Algorithm 1 is a subroutine for Algorithm 2 that removes each ABox axiom $D(x)$ for a concept D subsumed by C as well as

Input: KB $\mathcal{K}$, Concept $C = C_1 \sqcap C_2 \sqcap \ldots \sqcap C_n$ with $n \geq 1$, Individual x such that $\mathcal{K} \models C(x)$
Output: Two lists of counterfactuals sorted by min-likeliness and mean-likeliness (see (Eqs 2-3))
Function `counterfactual_candidates(`$\mathcal{K}$, C, x`):`
    $candidates \leftarrow [\,]$
    for ***each*** C_i **do**
        $\mathcal{K}_i \leftarrow$ `copy(`$\mathcal{K}$`)`
        `// Get copy of KB` $\mathcal{K}$
        $\mathcal{K}_i \leftarrow$ `create_candidates_neg(`$\mathcal{K}_i$, C_i, x`)`
        $candidates \leftarrow candidates \cup \{\mathcal{K}_i\}$
    end
    $cfs \leftarrow \underset{\mathcal{K}_i \in candidates}{\arg\min} \; \delta_{cf}(\mathcal{K}, \mathcal{K}_i)$
    $cfs\_min \leftarrow$ sort cfs by l_{min}
    $cfs\_mean \leftarrow$ sort cfs by l_{mean}
return cfs_min, cfs_mean

Algorithm 2: Generates counterfactual KBs $\mathcal{K}_i$ such that $\mathcal{K}_i \models C(x)$

Input: KB $\mathcal{K}$, Concept C, Individual x such that $\mathcal{K} \not\models C(x)$
Output: KB $\mathcal{K}$ such that $\mathcal{K} \not\models C(x)$
Function `create_candidates_pos(`$\mathcal{K}$, C, x`):`
    $C \leftarrow \{C_1, \ldots, C_n\}$
    for C_j *in* C **do**
        if C_j *is an existential restriction* $\exists r.D$ **then**
            add a new individual y
            add $r(x, y)$ to $\mathcal{K}$
            $\mathcal{K} \leftarrow$ `create_candidates_pos(`$\mathcal{K}$, D, y`)` `// recursive call`
        else
            add $C_j(x)$ to $\mathcal{K}$
    end
return $\mathcal{K}$

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

axioms $r(x, y)$ if $C = \exists r.A$, (with A any concept), thereby eliminating all possible ways of inferring $\mathcal{K} \models C(x)$. Once all the candidates (updated KBs) have been enumerated, the algorithm selects the KBs with the least edit distance as counterfactuals according to Eq 1. Furthermore, these counterfactuals are sorted by two likeliness measures (Eqs 2-3).

Theorem 1. *The Algorithm 2 is sound and complete: Given a KB* $\mathcal{K}$ *and an update request* $P = \langle C(x), rem \rangle$*, such that* $\mathcal{K} \not\vdash P$*, Algorithm 2 returns a collection of 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. Notice first that, given a KB $\mathcal{K}$ and an assertion $C(x)$ without intersections, Algorithm 1 yields an updated KB $\mathcal{K}'$ by essentially removing all possible ways of inferring $\mathcal{K} \models C(x)$. Algorithm 2 creates a copy of the KB for each C_i

intersected in C and calls Algorithm 1 as a subroutine for each C_i and KB K_i resulting in a list of candidates, which are then compared for their edit distances and likelihood measures.

To establish the correctness, we first prove that $\mathcal{K}' \vdash P$ is true for every KB $\mathcal{K}'$ returned by Algorithm 2. Since $\mathcal{K}' \in KBs$, there exists some $C_i \in C$ such that $\mathcal{K}'$ is obtained from $\mathcal{K}$ after applying changes due to some concept C_i and consequently $\mathcal{K}' = \mathcal{K}_i$. Moreover, the candidate $\mathcal{K}'$ is returned after the call `create_candidates_neg`$(\mathcal{K}, \{C_i\}, x)$ is made for some concept $C_i \in C$. This implies that there is some $C_i \in C$ such that $\mathcal{K}' \not\models C_i(x)$ since all the assertions necessary to infer $\mathcal{K} \models C_i(x)$ have been removed (Alg. 1). Consequently, $\mathcal{K}' \vdash P$ is true for each $\mathcal{K}' \in KBs$. Notice that, Line 9 in Algorithm 1 also removes assertions $r'(x, a)$ for a role r' such that $r' \sqsubseteq r$. Moreover, $\mathcal{K}'$ is non-redundant since exactly one candidate concept is chosen when C is conjunctive and local since Algorithm 1 only removes assertions containing x. This proves that the KBs returned by Algorithm 2 are indeed local counterfactuals and each of them fulfills the request P. Finally, Lines 10–12 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, $\mathcal{L} \not\models C_i(x)$ for at least one $C_i \in C$. We prove that $\mathcal{L} \in KBs$. Let $\mathcal{K}_i$ denote the candidate returned by the call `create_candidates_neg`$(\mathcal{K}, \{C_i\}, x)$. Clearly, $\mathcal{K}_i$ is obtained by removing all possible ways of inferring $\mathcal{K} \models C_i(x)$ from $\mathcal{K}$. Since $\mathcal{L} \not\models C_i(x)$ and $\mathcal{L}$ is non-redundant, it must be the case that $(\mathcal{K} \setminus \mathcal{K}_i) \subseteq (\mathcal{K} \setminus \mathcal{L})$, that is, the assertions removed to obtain $\mathcal{L}$ from $\mathcal{K}$ are also removed to obtain $\mathcal{K}_i$. As a result, we have that $\mathcal{L} \subseteq \mathcal{K}_i$. Now, suppose, to the contrary, that $\mathcal{K}_i \not\subseteq \mathcal{L}$. Since adding assertions is redundant, we assume without loss of generality that there is some assertion removed from $\mathcal{K}$ to obtain $\mathcal{L}$ which $\mathcal{K}_i$ still contains. But this implies that either the update $\mathcal{K} \rightarrow \mathcal{L}$ is redundant (if these additional removals concern assertions for concepts not subsumed by C) or $\mathcal{L}$ is not a local counterfactual candidate for P (if these concern assertions for some individual $y \neq x$). This is due to the fact that the removal of assertions for x resulting in $\mathcal{K}_i$ suffices to create a KB that fulfills P. This implies that $\mathcal{K}_i \subseteq \mathcal{L}$ and hence $\mathcal{L} = \mathcal{K}_i$. Consequently, $\mathcal{L}$ is returned in the iteration i when the subroutine is called for $C_i \in C$.

This completes the correctness in this direction and establishes the proof. □

Updating $\mathcal{K}$ to fulfill a request $\langle C(x), add \rangle$ is 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 CF for $C(x)$ in $\mathcal{K}$. However, our approach differs from abduction in that we also allow the addition of new 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 (starting with being part of no assertions except $\top(y)$) 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 new individual y. This new individual basically serves as a placeholder for any real-world individual with the needed features. This achieves the desired goal since our main focus is on finding CEs for $C(x)$. In practice, the decision to allow new 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 2. *The Algorithm 3 is sound and complete: Given a KB $\mathcal{K}$ and a request $P = \langle C(x), add \rangle$ such that $\mathcal{K} \not\models P$, 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 6–11 have been executed for each $C_j \in C$. This implies that $\mathcal{K}' \models C(x)$. Moreover, the applied changes are non-redundant and minimal. 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 1 implies that the update $\mathcal{K}' \to \mathcal{L}$ is redundant.

Materialization 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 us to choose that $\mathcal{A}$ should be fully materialized. The rationale behind materialization is the following: 1) all "features" (as authors in [14] 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.

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

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

our algorithm will get positive ratings by study participants. We used modified versions of the *Family* and *Animals* ontologies [45] 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 [11] with ELTL—the $\mathcal{EL}$ Tree Learner [10]—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 [11] 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 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. Using Wilcoxon signed-rank test for matched pairs (5.3 with alpha-level .01, this provides us with a power of .96 for a high effect size of .05 and .56 for a medium effect size of .03. 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.

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 Friedman 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.

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

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

scenario. Details on explanations selected by algorithm and participants can be found in our repository [1].

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 Study on Use Cases

To assess possible use cases for CEs in DL, we conducted an explorative study. The study was preregistered at https://osf.io/gazrq. We wanted to know about users' preferences for concept-based and/or counterfactual explanations, and how different use cases or concept length affect these. The full study material and data can be found at https://github.com/LNSieger/Counterfactual-Explanations-DL-ELH.

6.1 Method

We conducted an online survey via SoSciSurvey in German. Recruiting and setup of the study were similar to the survey. This time, participants were confronted with 4 different scenario stories. The first two were similar to the ones in the survey (based on the Family and Animals ontologies), the others were fictional stories not based on existing ontologies. One was about an AI helping select the right medicine for a fictional person and in which case another medicament would be the better choice. The other was about an AI deciding that the customer, called Clara, does not get a loan from the bank and what she could do to change

this. In this study, participants were both presented an explanation based on a concept (e.g. "I classify Petra as a mother, because she is female and has a child") and a CE to rate. For each scenario except the family scenario, participants were randomly assigned to one of two groups (between-subjects design): For one, a long concept (size >3) was used for explanation. The second group was presented with shorter concepts. Participants were asked to rate each information on a scale from 1-7 using three items: helpfulness for understanding the AI, usefulness, and if the information enhances controllability of the AI.

6.2 Results

In this section, we present the results of our explorative study on explanation use cases.

Sample 96 people took part in the survey. For the between-subjects questions, the resulting sample sizes were 48/48 or 47/49. Thus, for Mann-Whitney U test, targeting effect sizes of .03 and alpha-level .05, the power was 0.41. Age ranged between 20 and 70, mean = 35.0, median = 32, standard deviation = 11.0. 46 participants were female, 46 male, 3 diverse, one left the question blank. 17 had professions related to IT, 33 worked in unrelated fields, 31 did not state it clearly enough to tell (e.g., student). Therefore, we refrained from doing tests taking profession into account.

Factors related to explanation rating For each scenario and item, we used the Mann-Whitney U test to test for differences in concept-based explanation ratings from concept length. None of the results was significant (p>.05). Note that this might be due to the low power achieved for this test. Comparing counterfactual explanations with either long or short concept-based explanations, we found some differences: In the loan scenario, the CE was rated significantly higher on the item "This information is useful for Clara" than the concept-based explanation, which was highly significant for the long concept (Z = -3.29, p = <.001, Effect size = .34) and mildly significant for the short (Z = -2.23, p = <.05, Effect size = .23). On the other hand, regarding the animal scenario, participants found the short concept-based explanation more helpful than the counterfactual for themselves to understand the decision of the program (Z = -2.04, p = <.05, Effect size = .21). To explore differences in ratings over scenarios, we used the Friedma test (also called Friedman ANOVA). We found that, again for the item about usefulness, the CE was significantly rated higher in the loan scenario than in the medicine or animal scenario ($\chi(3)$ = 25.10, p <.0001) and also higher than in the family scenario. However, the latter difference was not significant by itself. Concept ratings were not affected by scenario. All explanations had high ratings (average ratings ranging between 5.13 and 6.39).

7 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, we used a rather restricted DL. Here, the KB is restricted to contain only non-complex concepts, i.e. without intersections or unions (though it should be noted that this is only due to algorithm 2, while 3 would still work if intersections are present). In other words, the concept we create a counterfactual from is an $\mathcal{ELH}$ concept, while the KB has the restriction to not contain intersections. Intersections could be supported with an approach similar to abductive reasoning [35,15,26] or axiom pinpointing [38,6,7]. However, in this first approach to counterfactual explanations in DL, we decided for a simple and fast algorithm for DLs with low expressiveness at first to lay the basis, and aim for a more exhaustive one supporting more expressive DLs in future work. Banning intersections (or unions) still fits a lot of application cases, since many ontologies only feature atomic subsumptions and assertions; for example, 9 out of 10 owl ontologies from the SML benchmark dataset collection [45] do not contain any intersections nor unions. We plan to expand our algorithm to the more complex DL $\mathcal{ALCH}$. The main challenges for $\mathcal{ALCH}$ include negation and disjunctions.

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 [44]. However, the Family ontology shows an example of cases where this is not possible, since people cannot usually change their gender or relatives. In this regard, it might also be interesting to refrain from the strategy to add new (fictive) individuals to the KB in some cases to make the concept hold. While this is an easy way to provide a general explanation, in some use cases, an algorithm that selects a fitting existing individual as possible object of a desired property could suggest even more concrete actions. Poyiadzi et al. [34] 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. 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 [30], but the learned concepts can become very long [28], 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 explainable AI, 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.

Concept-based vs. counterfactual explanations The explorative study gave us insight into people's thoughts regarding concept-based versus counterfactual explanations. While participants rated all explanations rather positively, we did not find much differences regarding the factors we took into account. However, regarding the loan scenario, we found that the counterfactual was rated more useful than the concept-based explanation. The loan and the medicine scenario differed from the other two, that were already used in the first survey, in the fact that the AI's classification directly affected a person. However, while the counterfactual was unclearly actionable (have a better cholesterol level) in the medicine scenario, the loan scenario's CE was designed to provide a concrete possible action (pay off debts) to change the classification. We suppose that this is the reason this explanation scored significantly higher on usefulness than the concept itself. Overall, our results suggest that counterfactuals perform at least as well as concept verbalization w.r.t. explaining algorithmic decisions. In addition, the study indicates that studying domains where counterfactual explanations lead to actionable decisions might be worthwhile.

8 Conclusion

We propose the first approach for generating CEs for concepts in DL. 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. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. Arrieta, A.B., Rodríguez, N.D., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion **58**, 82–115 (2020)

3. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: Dbpedia: A nucleus for a web of open data. In: ISWC/ASWC. LNCS, vol. 4825, pp. 722–735. Springer (2007)
4. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
5. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press (2017)
6. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic EL. In: Calvanese, D., Franconi, E., Haarslev, V., Lembo, D., Motik, B., Turhan, A., Tessaris, S. (eds.) Proceedings of the International Workshop on Description Logics (DL 2007). CEUR Workshop Proceedings, vol. 250. CEUR-WS.org (2007)
7. Baader, F., Peñaloza, R., Suntisrivaraporn, B.: Pinpointing in the description logic EL^+. In: Hertzberg, J., Beetz, M., Englert, R. (eds.) KI 2007: Advances in Artificial Intelligence, 30th Annual German Conference on AI, KI 2007, Proceedings. Lecture Notes in Computer Science, vol. 4667, pp. 52–67. Springer (2007)
8. Blanco, R., Ottaviano, G., Meij, E.: Fast and space-efficient entity linking for queries. In: WSDM. pp. 179–188. ACM (2015)
9. Brandt, S.: Reasoning in elh wrt general concept inclusion axioms. Institute for Theoretical Computer Science, Dresden University of Technology (2004)
10. Bühmann, L., Fleischhacker, D., Lehmann, J., Melo, A., Völker, J.: Inductive lexical learning of class expressions. In: EKAW. LNCS, vol. 8876, pp. 42–53. Springer (2014)
11. Bühmann, L., Lehmann, J., Westphal, P.: Dl-learner - A framework for inductive learning on the semantic web. J. Web Semant. **39**, 15–24 (2016)
12. Chai, X.: Diagnosis method of thyroid disease combining knowledge graph and deep learning. IEEE Access **8**, 149787–149795 (2020)
13. Chen, J., Chen, X., Horrocks, I., Myklebust, E.B., Jiménez-Ruiz, E.: Correcting knowledge base assertions. In: WWW. pp. 1537–1547. ACM / IW3C2 (2020)
14. Dandl, S., Molnar, C., Binder, M., Bischl, B.: Multi-objective counterfactual explanations. In: PPSN (1). LNCS, vol. 12269, pp. 448–469. Springer (2020)
15. Del-Pinto, W., Schmidt, R.A.: 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. pp. 2768–2775 (2019)
16. Demir, C., Ngomo, A.N.: DRILL- deep reinforcement learning for refinement operators in ALC. CoRR **abs/2106.15373** (2021)
17. Dervakos, E., Thomas, K., Filandrianos, G., Stamou, G.: Choose your data wisely: A framework for semantic counterfactuals. In: Elkind, E. (ed.) Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23. pp. 382–390. International Joint Conferences on Artificial Intelligence Organization (8 2023). https://doi.org/10.24963/ijcai.2023/43, https://doi.org/10.24963/ijcai.2023/43, main Track
18. Filandrianos, G., Thomas, K., Dervakos, E., Stamou, G.: Conceptual edits as counterfactual explanations. In: Martin, A., Hinkelmann, K., Fill, H., Gerber, A., Lenat, D., Stolle, R., van Harmelen, F. (eds.) Proceedings of the AAAI 2022 Spring Symposium on Machine Learning and Knowledge Engineering for Hybrid Intelligence (AAAI-MAKE 2022). CEUR Workshop Proceedings, vol. 3121. CEUR-WS.org (2022), https://ceur-ws.org/Vol-3121/paper6.pdf

19. Galárraga, L., Razniewski, S., Amarilli, A., Suchanek, F.M.: Predicting completeness in knowledge bases. In: WSDM. pp. 375–383. ACM (2017)
20. Gangemi, A., Nuzzolese, A.G., Presutti, V., Draicchio, F., Musetti, A., Ciancarini, P.: Automatic typing of dbpedia entities. In: ISWC (1). LNCS, vol. 7649, pp. 65–81. Springer (2012)
21. Grau, B., Ligozat, A.: A corpus for hybrid question answering systems. In: WWW (Companion Volume). pp. 1081–1086. ACM (2018)
22. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5), 93:1–93:42 (2019)
23. Heindorf, S., Blübaum, L., Düsterhus, N., Werner, T., Golani, V.N., Demir, C., Ngomo, A.N.: Evolearner: Learning description logics with evolutionary algorithms. In: WWW. pp. 818–828. ACM (2022)
24. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. In: NeurIPS (2020)
25. Iannone, L., Palmisano, I., Fanizzi, N.: An algorithm based on counterfactuals for concept learning in the semantic web. Appl. Intell. **26**(2), 139–159 (2007). https://doi.org/10.1007/s10489-006-0011-5, https://doi.org/10.1007/s10489-006-0011-5
26. Koopmann, P.: 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. pp. 1929–1935 (2021)
27. Kouagou, N.J., Heindorf, S., Demir, C., Ngomo, A.N.: Neural class expression synthesis. CoRR **abs/2111.08486** (2021)
28. Kouagou, N.J., Heindorf, S., Demir, C., Ngomo, A.N.: Learning concept lengths accelerates concept learning in ALC. In: ESWC. LNCS, vol. 13261, pp. 236–252. Springer (2022)
29. Lajus, J., Suchanek, F.M.: Are all people married?: Determining obligatory attributes in knowledge bases. In: WWW. pp. 1115–1124. ACM (2018)
30. Lehmann, J., Hitzler, P.: Concept learning in description logics using refinement operators. Mach. Learn. **78**(1-2), 203–250 (2010)
31. Meske, C., Bunde, E., Schneider, J., Gersch, M.: Explainable artificial intelligence: objectives, stakeholders, and future research opportunities. Information Systems Management **39**(1), 53–63 (2022)
32. Molnar, C.: Interpretable machine learning. Lulu. com (2020), christophm.github.io/interpretable-ml-book/
33. Negreanu, C., Karaoglu, A., Williams, J., Chen, S., Fabian, D., Gordon, A.D., Lin, C.: Rows from many sources: Enriching row completions from wikidata with a pretrained language model. In: WWW (Companion Volume). pp. 1272–1280. ACM (2022)
34. Poyiadzi, R., Sokol, K., Santos-Rodríguez, R., Bie, T.D., Flach, P.A.: FACE: feasible and actionable counterfactual explanations. In: AIES. pp. 344–350. ACM (2020)
35. Pukancová, J., Homola, M.: 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 (2018), https://ceur-ws.org/Vol-2211/paper-31.pdf
36. Rudnik, C., Ehrhart, T., Ferret, O., Teyssou, D., Troncy, R., Tannier, X.: Searching news articles using an event knowledge graph leveraged by wikidata. In: WWW (Companion Volume). pp. 1232–1239. ACM (2019)

37. Sarker, M.K., Hitzler, P.: Efficient concept induction for description logics. In: AAAI. pp. 3036–3043. AAAI Press (2019)
38. Schlobach, S., Cornet, R.: 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. pp. 355–362. Morgan Kaufmann (2003), http://ijcai.org/Proceedings/03/Papers/053.pdf
39. Stepin, I., Alonso, J.M., Catalá, A., Pereira-Fariña, M.: A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. IEEE Access **9**, 11974–12001 (2021)
40. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: WWW. pp. 697–706. ACM (2007)
41. Szklarczyk, D., Gable, A.L., Lyon, D., Junge, A., Wyder, S., Huerta-Cepas, J., Simonovic, M., Doncheva, N.T., Morris, J.H., Bork, P., Jensen, L.J., von Mering, C.: STRING v11: protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. Nucleic Acids Res. **47**(Database-Issue), D607–D613 (2019)
42. Trinh, T.T.Q., Chung, Y.C., Kuo, R.: A domain adaptation approach for resume classification using graph attention networks and natural language processing. Knowledge-Based Systems **266**, 110364 (2023)
43. Vrandecic, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
44. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the GDPR. Harv. JL & Tech. **31**, 841 (2018)
45. Westphal, P., Bühmann, L., Bin, S., Jabeen, H., Lehmann, J.: Sml-bench - A benchmarking framework for structured machine learning. Semantic Web **10**(2), 231–245 (2019)
46. Wishart, D.S., Feunang, Y.D., Guo, A.C., Lo, E.J., Marcu, A., Grant, J.R., Sajed, T., Johnson, D., Li, C., Sayeeda, Z., Assempour, N., Iynkkaran, I., Liu, Y., Maciejewski, A., Gale, N., Wilson, A., Chin, L., Cummings, R., Le, D., Pon, A., Knox, C., Wilson, M.: Drugbank 5.0: a major update to the drugbank database for 2018. Nucleic Acids Res. **46**(Database-Issue), D1074–D1082 (2018)
47. Zahera, H.M., Heindorf, S., Ngomo, A.N.: ASSET: A semi-supervised approach for entity typing in knowledge graphs. In: K-CAP. pp. 261–264. ACM (2021)
48. Zhan, Q., Yin, H.: 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. pp. 111–115 (2018)