On the Usefulness of Different Expert Question Types for Fault Localization in Ontologies

Patrick Rodler[0000−0001−8178−3333] and Michael Eichholzer

Alpen-Adria Universität Klagenfurt, 9020 Klagenfurt, Austria
patrick.rodler@aau.at
michael.eichholzer@aon.at

Abstract. When ontologies reach a certain size and complexity, faults such as inconsistencies or wrong entailments are hardly avoidable. Locating the faulty axioms that cause these faults is a hard and time-consuming task. Addressing this issue, several techniques for semi-automatic fault localization in ontologies have been proposed. Often, these approaches involve a human expert who provides answers to system-generated questions about the intended (correct) ontology in order to reduce the possible fault locations. To suggest as few and as informative questions as possible, existing methods draw on various algorithmic optimizations as well as heuristics. However, these computations are often based on certain assumptions about the interacting user and the metric to be optimized.

In this work, we critically discuss these optimization criteria and suppositions about the user. As a result, we suggest an alternative, arguably more realistic metric to measure the expert's effort and show that existing approaches do not achieve optimal efficiency in terms of this metric. Moreover, we detect that significant differences regarding user interaction costs arise if the assumptions made by existing works do not hold. As a remedy, we suggest a new notion of expert question that does not rely on any assumptions about the user's way of answering. Experiments on faulty real-world ontologies testify that the new querying method minimizes the necessary expert consultations in the majority of cases and reduces the computation time for the best next question by at least 80 % in all scenarios.

Keywords: Ontology Debugging · Interactive Debugging · Fault Localization · Sequential Diagnosis · Expert Questions · Ontology Quality Assurance · Ontology Repair · Test-Driven Debugging

1 Introduction

As Semantic Web technologies have become widely adopted in, e.g., government, security and health applications, the quality assurance of the data, information and knowledge used by these applications is a critical requirement. At the core of these semantic technologies, ontologies are a means to represent knowledge in a formal, structured and human-readable way, with a well-defined semantics. As ontologies are often developed and cured in a collaborative way by numerous contributors [34, 35], are merged by automated alignment tools [11], reach vast sizes and complexities [5], or use expressive logical formalisms such as OWL 2 [6], faults occur regularly during the evolution of ontologies [3, 11, 14, 29]. Since one of the major benefits of ontologies is the capability of using them to perform logical reasoning and thereby solve relevant problems, faults that affect the ontology's semantics are of particular concern for semantic applications. Specifically, such faults may cause the ontology, e.g., to become inconsistent, include unsatisfiable classes, or feature wrong entailments.

One important step towards the repair of such faults is the *localization* of the responsible faulty ontology axioms. To handle nowadays ontologies with often thousands of axioms, several fault localization approaches [9, 11, 12, 30] have been proposed to semi-automatically assist humans in this complex and time-consuming task, amongst them a plug-in, called ONTODEBUG[1] [27], for the popular ontology editor PROTÉGÉ. These approaches, which are mainly based on the *model-based diagnosis* framework [10, 15], use the faulty ontology along with additional specifications to reason about different fault assumptions. Such fault assumptions are called *diagnoses* if they are consistent with all given specifications. The specifications usually comprehend some requirements to the correct ontology, e.g., in the form of *logical properties* (e.g., consistency, coherency), and/or in terms of necessary and forbidden entailments. The latter are usually referred to as *positive and negative test cases* [4, 28, 30].

Research on model-based diagnosis has brought up various algorithms [9–11, 15, 16, 32] for computing and ranking diagnoses; however, a frequent problem is that a high number of competing diagnoses might exist where all of them lead to repaired ontologies with necessarily different semantics [16]. Finding the correct diagnosis (pinpointing the actually faulty axioms) is thus crucial for successful and sustainable repair. But, it is a mentally-demanding task for humans since it requires them to reason about and recognize entailments and non-entailments [7] of the ontology under particular fault assumptions. To relieve the user as much as possible, interactive techniques [16, 30] have been developed to undertake this task for the most part. What remains to be accomplished by the interacting human—usually an ontology engineer or a domain expert (referred to as *expert* in the sequel)—is the answering of a sequence of system-generated *queries* about the intended ontology. Roughly speaking, this involves the classification of certain axioms as either intended entailments (positive test cases) or non-intended entailments (negative test cases). Several evaluations [21, 24, 30, 32] have shown the feasibility and usefulness of such a query-based approach for fault localization, and its efficiency has been improved by various algorithmic optimizations [8, 20, 25, 31] and the use of heuristics [17, 18, 23, 26, 30] for the selection of the most informative questions to ask an expert.

However, the used heuristics, algorithms and optimization criteria are based on certain assumptions about the question answering behavior of experts. In this work, we critically discuss existing approaches with regard to these assumptions. Particularly, we characterize different types of experts and show that not all of them are equally well accommodated by current querying approaches. That is, we observe that the necessary expert interaction cost to locate the ontology's faults is significantly influenced by the way queries posed by the debugging system are answered. To overcome this issue, we propose a new way of user interaction that serves all discussed expert types equally well and moreover increases the expected amount of information relevant for fault localization obtained from the expert per asked axiom.

[1] All information about ONTODEBUG can be found at http://isbi.aau.at/ontodebug/

The main idea behind the new approach is to restrict queries—which are, for quite natural reasons, *sets of* axioms in existing methods—only to *single* axioms, as usually done in *sequential diagnosis* applications [10, 33], where systems different from ontologies (e.g., digital circuits) are analyzed and such singleton queries are the natural choice. That is, experts are asked single axioms at a time instead of getting batch queries which (possibly) include multiple axioms. Experiments on real-world faulty ontologies manifest the reasonability of the new approach. Specifically, in two thirds of the studied cases, the new querying technique is superior to existing ones in terms of minimizing the number of required expert inputs, regardless of the type of expert. In addition, the time for the determination of the best next query is reduced by at least $80\,\%$ in all investigated cases when using singleton queries instead of existing techniques.

The rest of the work is organized as follows. In Sec. 2, we give a short introduction to query-based fault localization in ontologies, before we challenge certain assumptions made by state-of-the-art approaches in the field in Sec. 3. We describe our proposed approach and discuss its pros and cons in Sec. 4. Our experiments and the obtained results are explicated in Sec. 5. We address research limitations and point to relevant future work topics in Sec. 6. In Sec. 7, we summarize the conclusions from this work.

2 Query-Based Fault Localization in Ontologies

We briefly recap basic technical concepts used in works on ontology fault localization, based on [16, 30]. As a running example we reuse the example presented in [23].

Fault Localization Problem Instance. We assume a faulty ontology to be given by the finite set of axioms $\mathcal{O} \cup \mathcal{B}$, where $\mathcal{O}$ includes the *possibly faulty* axioms and $\mathcal{B}$ the *correct* (background knowledge) axioms, and $\mathcal{O} \cap \mathcal{B} = \emptyset$ holds. This partitioning of the ontology means that faulty axioms must be sought only in $\mathcal{O}$, whereas $\mathcal{B}$ provides the fault localization context. At this, $\mathcal{B}$ can be useful to achieve a fault search space restriction (if parts of the faulty ontology are marked correct) or a higher fault detection rate (if external approved knowledge is taken into account, which may point at otherwise undetected faults). Besides logical properties such as consistency and coherency[2], requirements to the intended (correct) ontology can be formulated as a set of test cases [4], analogously as it is common practice in software engineering [2]. In particular, we distinguish between two types of test cases, positive (set P) and negative (set N) ones. Each test case is a set (interpreted as conjunction) of axioms; positive ones $p \in P$ *must* be and negative ones $n \in N$ *must not* be entailed by the intended ontology. We call $\langle \mathcal{O}, \mathcal{B}, P, N \rangle$ an *(ontology) fault localization problem instance (FPI).*

Example 1. Consider the following ontology with the terminology $\mathcal{T}$:[3]

$$\begin{aligned} \{ \quad & ax_1 : \mathit{ActiveResearcher} \sqsubseteq \exists \mathit{writes}.(\mathit{Paper} \sqcup \mathit{Review})\ , \\ & ax_2 : \exists \mathit{writes}.\top \sqsubseteq \mathit{Author}\ , \qquad ax_3 : \mathit{Author} \sqsubseteq \mathit{Employee} \sqcap \mathit{Person} \quad \} \end{aligned}$$

[2] An ontology $\mathcal{O}$ is *coherent* iff there do not exist any unsatisfiable classes in $\mathcal{O}$. A class C is *unsatisfiable* in $\mathcal{O}$ iff $\mathcal{O} \models C \sqsubseteq \bot$. See also [13, Def. 1 and 2].

[3] Throughout the presented examples, we use Description Logic notation. For details, see [1].

and assertions $\mathcal{A} : \{ax_4 : \mathit{ActiveResearcher}(ann)\}$. In natural language, the terminological axioms say that "an active researcher writes something which is a paper, a review or both" (ax_1), that "everybody who writes something is an author" (ax_2), and that "an author is both an employee and a person" (ax_3). To locate faults in the terminology while accepting as correct the assertion (ax_4) and stipulating that Ann is not necessarily an employee (negative test case $n_1 : \{\mathit{Employee}(ann)\}$), one can specify the following FPI: $\mathit{fpi}_{ex} := \langle \mathcal{T}, \mathcal{A}, \emptyset, \{n_1\}\rangle$. □

Fault Hypotheses. Let U_P denote the union of all positive test cases $p \in P$ and $\mathbf{C}_\bot := \{C \sqsubseteq \bot \mid C \text{ named class in } \mathcal{O}, \mathcal{B} \text{ or } P\}$. Given that the ontology, along with the positive test cases, is inconsistent or incoherent, i.e., $\mathcal{O} \cup \mathcal{B} \cup U_P \models x$ for some $x \in \{\bot\} \cup \mathbf{C}_\bot$, or some negative test case is entailed, i.e., $\mathcal{O} \cup \mathcal{B} \cup U_P \models n$ for some $n \in N$, some axioms in $\mathcal{O}$ must be accordingly modified or deleted to enable the formulation of the intended ontology. We call such a set of axioms $\mathcal{D} \subseteq \mathcal{O}$ a *diagnosis* for the FPI $\langle \mathcal{O}, \mathcal{B}, P, N\rangle$ iff $(\mathcal{O} \setminus \mathcal{D}) \cup \mathcal{B} \cup U_P \not\models x$ for all $x \in N \cup \{\bot\} \cup \mathbf{C}_\bot$. $\mathcal{D}$ is a *minimal diagnosis* iff there is no diagnosis $\mathcal{D}' \subset \mathcal{D}$. We call $\mathcal{D}^*$ *the actual diagnosis* iff all $ax \in \mathcal{D}^*$ are faulty and all $ax \in \mathcal{O} \setminus \mathcal{D}^*$ are correct. For efficiency and to point to minimally-invasive ontology repairs, fault localization approaches usually restrict their focus to the computation of minimal diagnoses.

Example 2. For $\mathit{fpi}_{ex} = \langle \mathcal{O}, \mathcal{B}, P, N\rangle$ from Example 1, $\mathcal{O} \cup \mathcal{B} \cup U_P$ entails the negative test case $n_1 \in N$, i.e., that Ann is an employee. The reason is that according to $ax_1 (\in \mathcal{O})$ and $ax_4 (\in \mathcal{B})$, Ann writes some paper or review since she is an active researcher. Due to the additional $ax_2 (\in \mathcal{O})$, Ann is also an author because she writes something. Finally, since Ann is an author, she must be both an employee and a person, as postulated by $ax_3 (\in \mathcal{O})$. Hence, $\mathcal{D}_1 : [ax_1]$, $\mathcal{D}_2 : [ax_2]$, $\mathcal{D}_3 : [ax_3]$ are (all the) minimal diagnoses for fpi_{ex}, as the deletion of any $ax_i \in \mathcal{O}$ breaks the unwanted entailment n_1. □

Eliminating Wrong Fault Hypotheses. The main idea model-based diagnosis systems use for fault localization—i.e., to find the actual diagnosis among the set of all (minimal) diagnoses—is that different fault assumptions have (necessarily [16]) different semantic properties in terms of entailments and non-entailments. This fact can be exploited to distinguish between diagnoses by posing queries to an expert. A query is a set of axioms Q which is entailed by some fault assumptions and inconsistent with some other fault assumptions. Asking a query Q corresponds to the question "Is (the conjunction of axioms in) Q an entailment of the intended ontology?". If answered positively, Q is added to the positive test cases P, and otherwise to the negative test cases N. The crucial property which makes a set of axioms Q a query is that at least one diagnosis is ruled out, regardless of whether Q is affirmed or negated. More formally:

Definition 1 (Query). *Given a set of minimal diagnoses* $\mathbf{D}$ *for an FPI* $\langle \mathcal{O}, \mathcal{B}, P, N\rangle$*, a set of axioms* Q *is a* query *(wrt.* $\mathbf{D}$*) iff at least one* $\mathcal{D}_i \in \mathbf{D}$ *is not a diagnosis for* $\langle \mathcal{O}, \mathcal{B}, P \cup \{Q\}, N\rangle$ *and at least one* $\mathcal{D}_j \in \mathbf{D}$ *is not a diagnosis for* $\langle \mathcal{O}, \mathcal{B}, P, N \cup \{Q\}\rangle$*.*

The expert who answers queries is modeled as a function $\mathsf{expert} : \mathbf{Q} \to \{y, n\}$ where $\mathbf{Q}$ is the query space; $\mathsf{expert}(Q) = y$ iff the answer to Q is positive, else $\mathsf{expert}(Q) = n$.

Example 3. Let the known set of diagnoses for fpi_{ex} be $\mathbf{D} = \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$ (see Example 2). One query wrt. $\mathbf{D}$ is, e.g., $Q_1 := \{\mathit{ActiveResearcher} \sqsubseteq \mathit{Author}\}$. Because, *(i)* adding Q_1 to P yields that the removal of $\mathcal{D}_1$ or $\mathcal{D}_2$ from $\mathcal{O}$ no longer breaks the unwanted entailment $\mathit{Employee}(\mathit{ann})$, i.e., $\mathcal{D}_1, \mathcal{D}_2$ are no longer minimal diagnoses, *(ii)* moving Q_1 to N means that $\mathcal{D}_3$ is not a minimal diagnosis anymore, as, to prevent the entailment of (the new negative test case) Q_1, at least one of ax_1, ax_2 must be deleted. Note, e.g., $Q_2 := \{\mathit{Author} \sqsubseteq \mathit{Person}\}$ is not a query since no diagnosis in $\mathbf{D}$ is invalidated upon assigning Q_2 to P, i.e., in case of a positive answer no useful information for diagnoses discrimination is gained. This is because Q_2 does not contribute to the violation of n_1 (in fact, the other "part" $\mathit{Author} \sqsubseteq \mathit{Employee}$ of ax_3 does so). □

Problem Definition. The query-based ontology fault localization problem (QFL) is to find for an FPI a series of queries to an expert, the answers of which lead to a single possible remaining fault assumption. The optimization version of the problem includes the additional goal to minimize the effort of the expert. Formally:

Problem 1 ((Optimal) QFL). **Given:** FPI $\langle \mathcal{O}, \mathcal{B}, P, N \rangle$. **Find:** (Minimal-cost) series of queries $Q_1, \dots, Q_k$ s.t. there is only one minimal diagnosis for $\langle \mathcal{O}, \mathcal{B}, P \cup P', N \cup N' \rangle$, where P' (N') is the set of all positively (negatively) answered queries, i.e., $P' := \{Q_i \mid 1 \leq i \leq k, \mathsf{expert}(Q_i) = y\}$ and $N' := \{Q_i \mid 1 \leq i \leq k, \mathsf{expert}(Q_i) = n\}$.

Note, there is no unified definition of the cost of a solution to the QFL problem. Basically, any function mapping the series $Q_1, \dots, Q_k$ to a non-negative real number is possible. We pick up on this discussion again in Sec. 3.

Example 4. Let the actual diagnosis for fpi_{ex} be $\mathcal{D}_3$, i.e., ax_3 is the (only) faulty axiom in $\mathcal{O}$ (intuition: an author is not necessarily employed, but might be, e.g., a freelancer). Then, given fpi_{ex} as an input, solutions to Problem 1, yielding the final diagnosis $\mathcal{D}_3$, are, e.g., $P' = \emptyset, N' = \{\{\exists \mathit{writes}.\top \sqsubseteq \mathit{Employee}\}, \{\mathit{Author} \sqsubseteq \mathit{Employee}\}\}$ or $P' = \{\{\mathit{ActiveResearcher} \sqsubseteq \mathit{Author}\}\}, N' = \emptyset$. Measuring the querying cost by the number of queries, the latter solution (cost: 1) is optimal, the former (cost: 2) not. □

Query-based Fault Localization. Given an FPI as input, the ontology fault localization process basically consists of four iteratively repeated steps: First, the fault hypotheses computation yielding a sample of diagnoses; second, the determination of the best next query based on the known diagnoses; third, the information acquisition where an expert answers the suggested query; and, fourth, the integration of the gathered information, involving the extension of the FPI's test cases based on the posed query and the given answer. The reiteration of these phases is continued until a stop criterion is met, e.g., a single diagnosis remains. This remaining diagnosis then provably contains only faulty axioms [16].[4] In the following, we will call one execution of this process starting with an input FPI until a single diagnosis is isolated a *fault localization session*.

[4] Note, the finally remaining diagnosis does not necessarily contain *all* faulty axioms in the ontology, as, e.g., some existing faults in the ontology might not yet have surfaced in terms of problems such as wrong entailments or unsatisfiable classes. However, the (faultiness of the) axioms in the final diagnosis do(es) explain *all observed problems* in the ontology.

3 Discussion of Query-based Fault Localization Approaches

In this section we analyze existing approaches regarding the assumptions they make about (the query answering behavior of) the interacting user, their properties resulting from natural design choices, as well as optimization criteria they consider.

Assumptions about Query Answering. All approaches that draw on the interactive methodology described in Sec. 2 make the assumption *during their computations and optimizations* that the expert evaluates each query as a whole. That is, they perform an assessment of the query effect or (information) gain *based on two possible outcomes* (y and n). However, in fact, since queries might contain multiple axioms, the feedback of an expert to a query might take a multitude of different shapes. Because, the expert might not view the query as an atomic question, but at the axiom level, i.e., inspecting axioms one-by-one. Clearly, to answer the query $Q = \{ax_1, \dots, ax_m\}$ positively—i.e., that the conjunction of the axioms $ax_1, \dots, ax_m$ is an entailment of the intended ontology—one needs to scrutinize and approve the entailment of all single axioms. To negate the query Q, in contrast, it suffices to detect one of the m axioms in Q which is not an entailment of the intended ontology. In this latter case, however, we might reasonably assume the interacting expert to be able to name (at least this) one *specific* axiom $ax^* \in Q$ that is not an intended entailment. We might think of ax^* as a "witness of the falsehood of the query". This additional information—beyond the mere negative answer n indicating that some *undefined* query-axiom must not be entailed—justifies the addition of $n^* := \{ax^*\}$, instead of Q, to the negative test cases. Please note that n^* provides stronger information than Q, and thus potentially rules out more diagnoses. The reason is that each diagnosis that entails Q (i.e., is invalidated given the negative test case Q) particularly entails ax^* (i.e., is definitely invalidated given the negative test case n^*). Apart from the scenario where experts provide just a falsehood-witness in the negative case, they might give even more information. For instance, an expert could walk through the query-axioms until either a non-entailed one is found or all axioms have been verified as intended entailments. In this case, there might as well be some entailed axioms encountered before the first non-entailed one is detected. The set of these entailed axioms could then be added to the positive test cases—in addition to the negative test case n^*. Alternatively, the expert might also continue evaluating axioms after recognizing the first non-entailed axiom ax^*, in this vein providing the classification of all single query-axioms in Q.

Based on this discussion, we might—besides the *query-based* expert that answers queries as a whole, exactly as specified by the expert function defined in Sec. 2—characterize (at least) three different types of *axiom-based* experts which supply information beyond the mere n label for a query Q in the negative case:[5]

- *Minimalist:* Provides exactly one $ax^* \in Q$ which is not entailed by the intended ontology.
- *Pragmatist:* Provides the first found axiom $ax^* \in Q$ that is not entailed by the intended ontology, and additionally all axioms evaluated as entailments of the intended ontology until ax^* was found.

[5] Note, a positive answer (y) implicitly provides *axiom-level* information, i.e., the positive classification of all query-axioms. Thus, the discussed experts differ only in their negation behavior.

- *Maximalist:* Provides the classification of each axiom in Q as either an entailment or a non-entailment of the intended ontology.

Consequently: *(i)* Without knowing the answering type of the interacting expert in advance, the binary query evaluation conducted in existing works is generally only an approximation. *(ii)* Even if the expert type is known, it is an open question which form of interaction, i.e., which way of asking the expert, allows to exploit the expert knowledge most beneficially and economically. Our experimental evaluations reported in Sec. 5 shall confirm (i) and bring light to (ii).

Natural Design Choices. As explicated in Sec. 2, the principle behind queries is the comparison of entailments and non-entailments resulting from different fault assumptions (diagnoses). In existing works [26, 30], this is often done by computing common entailments for some diagnoses and verifying whether assuming correct these entailed axioms leads to an inconsistency with some other diagnosis. In the light of this strategy, it is quite natural to specify queries as *sets of* axioms. The reasons are the following:

First, it stands to reason to use and further process *all* entailments that a reasoner outputs. Second, the fewer entailments are used, the higher is the chance that these are entailed by all (known) diagnoses and hence do not constitute a query. In fact, it has been shown in [24] that such unsuccessful query verifications can account for a massive query computation time overhead. Third, allowing queries to include a larger number of axioms implies a larger query search space and thus enables to identify a better next query—where "better" applies to the case where a *query-based* expert is assumed and query selection heuristics [23] are used that aim at minimizing the *number of queries*.

Optimization Criteria. The meaning of "minimal-cost" in Problem 1 might be defined in different ways. Most existing works on query-based fault localization, e.g., [16, 26, 27, 30], specify the cost of a solution $Q_1, \dots, Q_k$ to the QFL problem to be *the number of* queries, i.e., k. The underlying assumption in this case is that any two queries mean the same (answering) cost for an expert. Given that queries might include fewer or more axioms of lower or higher (syntactic or semantic) complexity, we argue that this cost measure might be too coarse-grained to capture the effort for an interacting expert in a realistic way. Instead, it might be more suitable to measure the costs at the axiom level.

However, there is a fundamental problem with the optimization criterion that aims at minimizing the number of query-axioms an expert needs to classify during an interactive fault localization session. Because, adopting this criterion, the evaluation and comparison of the goodness of queries while searching for the best next query trivially requires the calculation of the specific query-axioms—for a potentially large number of query candidates. However, the calculation of the specific query-axioms is generally costly in that it involves a high number of calls to expensive reasoning services. A remedy to this problem in terms of a two-staged technique which (i) can assess queries without knowing the specific axioms they contain and (ii) minimizes both the number of queries and the costs at the axiom level is suggested by [24]. However, the expert type taken as a basis for these optimizations is again the *query-based* one, and the number of axioms is only the secondary minimization criterion after the number of queries.

4 New Approach to Expert Interaction

Idea. In the light of the issues pointed out in Sec. 3 and following quite straightforward from the given argumentation, we propose a new way of expert interaction for fault localization in ontologies, namely to abandon "batch-queries" including multiple axioms and to focus on so-called *singleton queries* instead. That is, we suggest to restrict queries to only single-axiom questions. Formally:

Definition 2 (Singleton Query). *Let* $\mathbf{D}$ *be a set of diagnoses for an FPI* $\langle \mathcal{O}, \mathcal{B}, P, N \rangle$*. Then,* Q *is a singleton query (wrt.* $\mathbf{D}$*) iff* Q *is a query (wrt.* $\mathbf{D}$*) and* $|Q| = 1$.[6]

Properties. The *advantages* of singleton queries are the following:

- *Maximally-fine granularity of optimization loop:* Each atomic expert input (i.e., each classified axiom) can be directly taken into account to optimize further computations and expert interactions. Simply put, each axiom the expert is asked to classify is a function of *all* so-far classified axioms.
- *Smaller search space:* There are fewer singleton queries than there are general queries. Therefore, the worst-case search costs for singleton queries are bounded by the worst-case search costs for normal queries.
- *Realistic query assessment:* For singleton queries, the binary-outcome assessment performed by existing approaches is exact, plausible and not just an approximation of the possible real cases, independent of the expert (type). The reason is that there *are* exactly two possible outcomes, namely y (query-axiom is an intended entailment) and n (query-axiom is a non-intended entailment).
- *Direct re-use of existing works:* Concepts (e.g., heuristics) and techniques (e.g., search algorithms) devised for queries can be immediately re-used for singleton queries, because each singleton query *is* a (specific) query.
- *Unequivocal optimization criterion:* Minimization of the number of queries and minimization of the number of query-axioms coincide for singleton queries. This unifies the two competing and arguable views on the query optimization problem.
- *More informative feedback per axiom (assuming query-based expert):* For both singleton and normal queries, the positive assessment of the query implies that all axioms in it are intended entailments. That is, the information acquired per axiom is equal. In case the query is negated, however, singleton queries generally provide more information per axiom. Because, for a normal query a negative answer corresponds to the information that *one of a set of* axioms is not true, whereas we learn from a negated singleton query that *one particular* axiom must not be entailed.
- *Same fault localization efficiency for all expert types:* Singleton queries, by their nature, admit only one style of answering—the answer is positive iff the single comprised axiom must be entailed by the intended ontology, and negative iff it must not be entailed. Thus, all discussed expert types coincide for singleton queries. As an implication of this, it is neither required to ascertain the expert type a priori nor to adapt algorithms to different experts, which makes the query optimization process simpler and the outcome equally suitable for all (discussed) types of users.

[6] To stress the difference between singleton queries (Def. 2) and queries in terms of Def. 1, we will henceforth often refer to the latter as *normal queries*.

On the downside, the smaller search space—apart from the advantage it brings regarding the worst-case query search complexity—can be seen as a *disadvantage* as well. The reason is that soundness of the query search is more difficult to obtain, i.e., more considerations and computations than for normal queries are required to ensure that the search outcome is indeed a *singleton* query (cf. the discussion on "Natural Design Choices" in Sec. 3). To tackle this, one could try to generate normal queries and post process them by means of query-size minimization techniques similar to those used by existing works [17, 30]. The problem is, however, that these techniques do not guarantee the reduction to a single axiom.

Thus, beside all the advantages of singleton queries, an algorithmic and computational challenge towards their efficient generation and optimization remains to be solved.

Computation and Optimization. Despite this open issue regarding general singleton queries, we were able to develop a polynomial time and space algorithm for singleton queries of the form $\{ax\}$ where $ax \in \mathcal{O}$.[7] This algorithm gets an FPI $\langle \mathcal{O}, \mathcal{B}, P, N \rangle$, a set of known minimal diagnoses $\mathbf{D}$ as well as a query selection heuristic h (among those discussed in [23]) as an input, and outputs the globally optimal singleton query of the above-mentioned form. At this, "globally optimal" means optimal in terms of h among *all* queries in the query space. The basis for our algorithm is provided by the theory and strategies for normal queries elaborated in [17], which we extended and adapted accordingly to obtain a method for singleton queries. The full description of the new algorithm is beyond the scope of this work and can be found in [19]. Here, we rather focus on understanding the added value of singleton queries and their comparison with normal queries.

5 Evaluation

Goal. The aim of the following experiments is the analysis of normal queries under different answering conditions (expert types discussed in Sec. 3) and the comparison between normal queries and the proposed singleton queries. Focus of the investigations is the *required effort for the expert* for fault localization and the *query computation time*.

Dataset, Experiment Settings and Measurements. The dataset of faulty (inconsistent and/or incoherent) real-world ontologies used in our experiments is given in Tab. 1. We used each of these ontologies $\mathcal{O}$ to specify an FPI as $fpi := \langle \mathcal{O}, \emptyset, \emptyset, \emptyset \rangle$, i.e., the background knowledge $\mathcal{B}$ as well as the positive (P) and negative (N) test cases were initially empty. Tab. 1 also gives an idea of the *diagnostic structure* of the considered FPIs, in terms of the size and logical expressivity[8] of the ontology, as well as the number and minimal/maximal size of all minimal diagnoses for the initial problem. As query selection heuristics (h) we used the measures discussed in [26, 30]. These are

[7] Such (singleton) queries consisting of only axioms explicitly included in the ontology are called *explicit* (singleton) *queries* [17].

[8] The logical expressivity refers to the power of the logical language used in the ontology in terms of how much can be expressed using this language. In general, the higher the expressivity, the higher the cost of reasoning (and thus the cost of computing queries) with the respective logic tends to be. See [1] for more details on the logical expressivity of ontologies.

Table 1: Dataset of faulty ontologies used in the experiments, sorted by the ontology size $|\mathcal{O}|$.

ontology $\mathcal{O}$	$\lvert\mathcal{O}\rvert$	expressivity 1)	#D/min/max 2)
Koala (K) 3)	42	$\mathcal{ALCON}^{(D)}$	10/1/3
University (U) 4)	50	$\mathcal{SOIN}^{(D)}$	90/3/4
MiniTambis (M) 4)	173	$\mathcal{ALCN}$	48/3/3
Transportation (T) 4)	1300	$\mathcal{ALCH}^{(D)}$	1782/6/9
Economy (E) 4)	1781	$\mathcal{ALCH}^{(D)}$	864/4/8
DBpedia (D) 5)	7228	$\mathcal{ALCHF}^{(D)}$	7/1/1

Key:
1): Description Logic expressivity [1].
2): #D, min, max denote the number, the minimal as well as the maximal size of minimal diagnoses for the input FPI.
3): Faulty ontology included in the Protégé Project.
4): Sufficiently complex FPIs (#D ≥ 40) used in [30].
5): Faulty version of the DB-Pedia ontology, downloaded from https://bit.ly/2RUVbMj.

ENT (maximize information gain per query), SPL (maximize worst-case diagnoses elimination rate per query) and RIO (optimize balance between ENT and SPL per query).

For each FPI and each heuristic h we ran 20 fault localization sessions, each time using a different randomly specified actual diagnosis $\mathcal{D}^*$ to be located. To automatically answer queries throughout a session in a way the predefined diagnosis $\mathcal{D}^*$ is finally located, we implemented a function based on $\mathcal{D}^*$ which simulates the interacting user. Specifically, the *query-based* expert was simulated by always outputting an answer to a query Q that does not effectuate the invalidation of $\mathcal{D}^*$; the *axiom-based* experts (*minimalist*, *pragmatist*, *maximalist*) were simulated in a way that, if they classify an axiom ax at all (cf. Sec. 3), then as an entailment if $ax \notin \mathcal{D}^*$ and as a non-entailment else. The size of the diagnoses sample generated before each query computation was set to $|\mathbf{D}| = 10$. Since two of the used heuristics (ENT, RIO) depend on diagnosis probabilities, we sampled and assigned uniform random probabilities to diagnoses for each FPI. For query generation throughout the fault localization sessions, we used the algorithms described in [17] (for normal queries) and [19] (for singleton queries). Note, all (normal and singleton) queries Q computed in our experiments were restricted to include axioms that occur in the ontology $\mathcal{O}$, i.e., $Q \subseteq \mathcal{O}$ for all queries Q.[9]

For each performed fault localization session we measured the number of answered queries (*#Q*) as well as the number of classified query-axioms (*#Ax*) required until the predefined $\mathcal{D}^*$ was found with certainty (i.e., until all other diagnoses were ruled out through the answered queries), and the average computation time to find the best next query (*time per Q*).

Experiment Results. First, we observe that, for normal queries, the answering style has a significant impact on the expert's effort, both when using #Ax and #Q as a cost metric. In fact, any axiom-based strategy (pragmatist, maximalist or minimalist) is better than a query-based one (bars in Fig. 1), with savings of up to 57 % wrt. #Ax and up to 58 % wrt. #Q (cf. ENT, pragmatist vs. query-based, M ontology, in Fig. 1). The reason for this is that an axiom-based approach involves strictly more informative answers than a query-based one (cf. Sec. 3).

Second, also among the axiom-based expert types, there are notable cost differences (wrt. #Ax). As it turns out, the pragmatist approach is clearly the best choice to answer normal queries for *all* investigated ontologies.[10] Also, when measuring the cost by #Q

[9] This is owed to the fact that the efficient generation of optimal singleton queries including "implicit" axioms, i.e., where $Q \not\subseteq \mathcal{O}$ holds, is still an open research topic (cf. Sec. 4).

[10] Note, the presented figures do not expose all results. However, the observations were greatly consistent over all studied ontologies. See the extended version [19] of this paper for all plots.

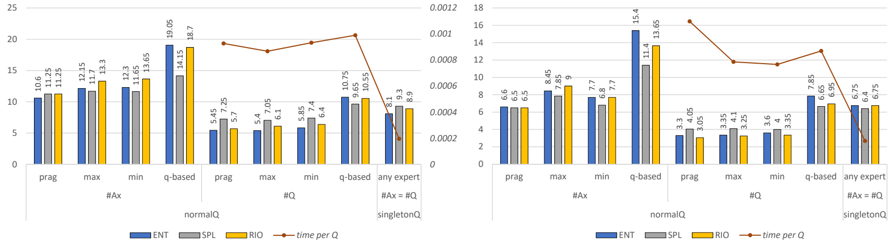

Fig. 1: **Overview of observations** for ontology E (left) and M (right): The bars show *#Ax* and *#Q* for heuristics ENT (blue), SPL (gray) and RIO (yellow) and for expert types *min*imalist, *prag*matist, *max*imalist, and *q*uery-*based* expert (cf. Sec. 3), for normal queries (*normalQ*) and singleton queries (*singletonQ*). The red line reports *time per Q* (in sec). All plotted values are averages over all 20 fault localization sessions. Bars refer to left y-axis, red line to right y-axis.

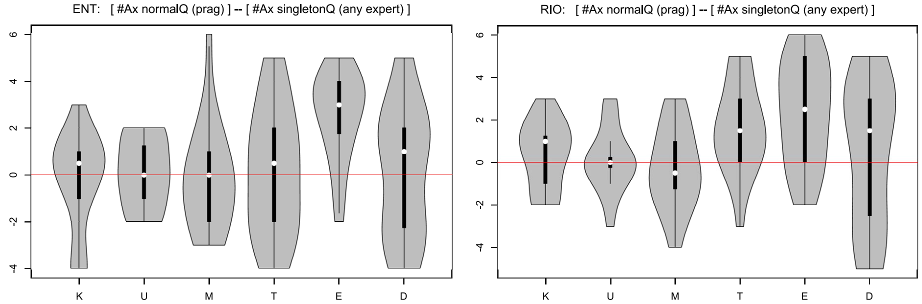

Fig. 2: **Comparison between normal and singleton queries** for ENT (left) and RIO (right) heuristics: The violin plots show the difference in query answering effort (*#Ax*) between using the best answering strategy (*prag*matist) for normal queries (*normalQ*) and using singleton queries (*singletonQ*), for all ontologies (x-axis) given in Tab. 1. Each violin plot summarizes the differences per session over all 20 fault localization sessions. White dots in plots indicate the median; if above/below zero (red line), singleton/normal queries are better in the majority of the sessions.

(as existing works do), the pragmatist tends to be the most reasonable type, albeit the differences are just marginal in this case. So far, we conclude that normal queries, for best efficiency and regardless of the adopted query selection heuristic, should *not* be answered simply by y or n, but the interacting expert should evaluate the individual query-axioms, pursuing the pragmatist method (cf. Sec. 3). Note, it is surprising that *one* (axiom-based) answering strategy *always* prevails, as normal queries are optimized based on the assumption of the (fairly different) query-based user.

Third, when comparing singleton with normal queries (answered by the pragmatist strategy), the costs wrt. #Ax are often pretty similar on average (see, e.g., M ontology in Fig. 1), even though with a notable tendency towards a superiority of singleton queries. E.g., for the E ontology and ENT heuristic, we measure an average effort overhead of more than 30 % when relying on normal queries as opposed to singleton ones (Fig. 1). Fig. 2 gives a clearer picture of this comparison. E.g., it reveals that, for all ontologies, singleton queries were at least as good as normal ones in the majority of sessions when using ENT as a heuristic. For the RIO heuristic, the results are similar, and in three cases (ontologies K, T, D) even more in favor of singleton queries than for ENT. Over all ontologies and heuristics, singleton queries even led to less expert interactions in more than 66 % of the sessions. However, there are scenarios where normal queries outperform singletons on average as well, as evidenced by the RIO and M ontology combination. Moreover, in most scenarios the proportion of sessions where normal queries mean fewer expert consultations (area of violin plots below the red line) is significant. Thus, normal queries *are* a reasonable way of expert interaction, but can match up to singleton queries only if the pragmatist answering behavior is given.

Regarding the computation time per query, we clearly recognize (red lines in Fig. 1) that (optimal) singleton queries are significantly faster determined than (optimal) normal queries. The savings *always* amounted to between 80 % and 90 %.

6 Research Limitations and Future Work

First, the evaluations in this work are based on simulations of fault localization sessions and objective measures such as computation times or the number of required queries. Although this objective assessment shows a higher average efficiency of the new approach as compared to existing ones, it is important to validate the subjective usefulness of the suggested querying technique, for instance in terms of a user study. This is part of our future work. However, it nevertheless stands to reason that users familiar with normal queries would likewise accept and adopt singleton queries, just because singleton queries represent a particularly *simple subclass of normal queries*.

A second limitation is the restriction to explicit queries—those that are constituted by axioms from the ontology at hand—in our empirical analyses. The reason we did so is because we currently only have an algorithm for the computation and optimization of explicit singleton queries, by drawing on and extending the theory elaborated in [17]. The finding of an *efficient* algorithm that soundly generates implicit singleton queries, in contrast, is an open issue and on our future work agenda. That said, as soon as we have developed an adequate algorithm, we plan to do similar evaluations as done in this work for singleton and normal queries without the restriction to explicit queries.

As a third limitation, it should be noted that the analyzed expert types, as discussed in Sec. 3, provide by no means a complete characterization of all possible cases that could arise. While the discussion in this work bases on the assumption that an expert will provide for each query at the minimum as much information as is necessary to classify the entire query as a positive or negative test case (cf. the expert function in Sec. 2), there are (at least) two further query answering scenarios that are worthwhile considering. First, there is the case where the expert classifies a proper subset (or even none) of the axioms of a normal query positively while not labeling any axiom negatively, e.g., due to laziness or lack of knowledge. Second, there is the case where an expert might misclassify axioms when answering queries. Such “oracle errors” were observed quite commonly in the studies conducted in [21]. Investigating these two scenarios for normal and singleton queries as well as the conception of strategies how to handle these cases is another research avenue we will prospectively pursue.

7 Conclusions

We critically discuss design choices, made assumptions and used optimization criteria of state-of-the-art query-based ontology fault localization approaches. Based on the revealed issues, we propose a new way of asking questions to an expert. Theoretical and empirical analyses using real-world problems demonstrate significant advantages of the novel querying method. Among other things, we learn that the suggested method—as opposed to existing approaches—(1) is simpler, (2) enables exact query optimizations instead of only approximate ones, (3) implies a more than 80 % reduction of the expert's waiting time for the next question, (4) enforces more informative expert inputs, (5) leads to the least fault localization effort for the expert in more than 66 % of the cases, and (6) guarantees the same efficiency regardless of the expert's (answering) behavior. Notably, our method is basically applicable to any monotonic knowledge representation language [16], as well as to other model-based diagnosis applications [22].

Acknowledgments

This work was in part supported by the Carinthian Science Fund (KWF), contract KWF-3520/26767/38701. Moreover, we thank Wolfgang Schmid for his technical support during the implementation of our experiments.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook. Cambridge University Press, 1st edn. (2003)
2. Beck, K.: Test-driven development: by example. Addison-Wesley Professional (2003)
3. Ceusters, W., Smith, B., Goldberg, L.: A terminological and ontological analysis of the NCI thesaurus. Methods of information in medicine **44**(4), 498 (2005)
4. Felfernig, A., Friedrich, G., Jannach, D., Stumptner, M.: Consistency-based diagnosis of configuration knowledge bases. Artif. Intell. **152**(2), 213–234 (2004)
5. Golbeck, J., Fragoso, G., Hartel, F., Hendler, J., Oberthaler, J., Parsia, B.: The national cancer institute's thesaurus and ontology. JWS **1**(1) (2003)
6. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. JWS **6**(4), 309–322 (2008)
7. Horridge, M., Bail, S., Parsia, B., Sattler, U.: The cognitive complexity of OWL justifications. In: ISWC. pp. 241–256. Springer (2011)
8. Jannach, D., Schmitz, T., Shchekotykhin, K.: Parallel model-based diagnosis on multi-core computers. JAIR **55**, 835–887 (2016)
9. Kalyanpur, A.: Debugging and Repair of OWL Ontologies. PhD-th, Univ. Maryland (2006)
10. de Kleer, J., Williams, B.C.: Diagnosing multiple faults. Artif. Intell. **32**(1), 97–130 (1987)
11. Meilicke, C.: Alignment incoherence in ontology matching. PhD-th, Univ. Mannheim (2011)
12. Nikitina, N., Rudolph, S., Glimm, B.: Interactive ontology revision. JWS **12**(0) (2012)
13. Qi, G., Hunter, A.: Measuring incoherence in description logic-based ontologies. In: ISWC. pp. 381–394. (2007)
14. Rector, A.L., Brandt, S., Schneider, T.: Getting the foot out of the pelvis: modeling problems affecting use of SNOMED CT hierarchies in practical applications. JAMIA **18**(4), (2011)
15. Reiter, R.: A Theory of Diagnosis from First Principles. Artif. Intell. **32**(1), 57–95 (1987)

16. Rodler, P.: Interactive Debugging of Knowledge Bases. PhD-th, Univ. Klagenfurt (2015)
17. Rodler, P.: Towards better response times and higher-quality queries in interactive KB debugging. Tech. rep., Univ. Klagenfurt (2016), http://arxiv.org/abs/1609.02584v2
18. Rodler, P.: On active learning strategies for sequential diagnosis. In: DX. pp. 264–283 (2018)
19. Rodler, P., Eichholzer, M.: A New Expert Questioning Approach to More Efficient Fault Localization in Ontologies. Tech. rep., Univ. Klagenfurt (2019), http://arxiv.org/abs/1904.00317
20. Rodler, P., Herold, M.: StaticHS: A variant of Reiter's hitting set tree for efficient sequential diagnosis. In: SoCS. pp. 72–80 (2018)
21. Rodler, P., Jannach, D., Schekotihin, K., Fleiss, P.: Are Query-Based Ontology Debuggers Really Helping Knowledge Engineers? CoRR abs/1904.01484 (2019), http://arxiv.org/abs/1904.01484
22. Rodler, P., Schekotihin, K.: Reducing model-based diagnosis to knowledge base debugging. In: DX. pp. 284–296 (2018)
23. Rodler, P., Schmid, W.: On the impact and proper use of heuristics in test-driven ontology debugging. In: RuleML+RR. pp. 164–184 (2018)
24. Rodler, P., Schmid, W., Schekotihin, K.: A generally applicable, highly scalable measurement computation and optimization approach to sequential model-based diagnosis. CoRR abs/1711.05508 (2017), http://arxiv.org/abs/1711.05508
25. Rodler, P., Schmid, W., Schekotihin, K.: Inexpensive cost-optimized measurement proposal for sequential model-based diagnosis. In: DX. pp. 200–218 (2018)
26. Rodler, P., Shchekotykhin, K., Fleiss, P., Friedrich, G.: RIO: Minimizing User Interaction in Ontology Debugging. In: RR. pp. 153–167 (2013)
27. Schekotihin, K., Rodler, P., Schmid, W.: Ontodebug: Interactive ontology debugging plug-in for Protégé. In: FoIKS. pp. 340–359 (2018)
28. Schekotihin, K., Rodler, P., Schmid, W., Horridge, M., Tudorache, T.: A Protégé plug-in for test-driven ontology development. In: ICBO. (2018)
29. Schulz, S., Schober, D., Tudose, I., Stenzhorn, H.: The pitfalls of thesaurus ontologization–the case of the NCI thesaurus. In: AMIA Annual Symposium. (2010)
30. Shchekotykhin, K., Friedrich, G., Fleiss, P., Rodler, P.: Interactive Ontology Debugging: Two Query Strategies for Efficient Fault Localization. JWS **12-13**, 88–103 (2012)
31. Shchekotykhin, K., Jannach, D., Schmitz, T.: Mergexplain: Fast computation of multiple conflicts for diagnosis. In: IJCAI. pp. 3221–3228 (2015)
32. Shchekotykhin, K.M., Friedrich, G., Rodler, P., Fleiss, P.: Sequential diagnosis of high cardinality faults in knowledge-bases by direct diagnosis generation. In: ECAI. (2014)
33. Siddiqi, S.A., Huang, J.: Sequential diagnosis by abstraction. In: JAIR **41**, 329–365 (2011)
34. Smith, B., Ashburner, M., Rosse, C., et al.: The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology **25**(11), (2007)
35. Tudorache, T., Noy, N.F., Tu, S., Musen, M.A.: Supporting collaborative ontology development in Protégé. In: ISWC. pp. 17–32 (2008)