On the Impact and Proper Use of Heuristics in Test-Driven Ontology Debugging

Patrick Rodler(✉) and Wolfgang Schmid

Alpen-Adria Universität Klagenfurt, 9020 Klagenfurt, Austria
{patrick.rodler,wolfgang.schmid}@aau.at

Abstract. Given an ontology that does not meet required properties such as consistency or the (non-)entailment of certain axioms, Ontology Debugging aims at identifying a set of axioms, called diagnosis, that must be properly modified or deleted in order to resolve the ontology's faults. As there are, in general, large numbers of competing diagnoses and the choice of each diagnosis leads to a repaired ontology with different semantics, Test-Driven Ontology Debugging (TOD) aims at narrowing the space of diagnoses until a single (highly probable) one is left. To this end, TOD techniques automatically generate a sequence of queries to an interacting oracle (domain expert) about (non-)entailments of the correct ontology. Diagnoses not consistent with the answers are discarded. To minimize debugging cost (oracle effort), various heuristics for selecting the best next query have been proposed. We report preliminary results of extensive ongoing experiments with a set of such heuristics on real-world debugging cases. In particular, we try to answer questions such as "Is some heuristic always superior to all others?", "On which factors does the (relative) performance of the particular heuristics depend?" or "Under which circumstances should I use which heuristic?".

Keywords: Ontology debugging · Test-driven debugging
Query selection · Heuristics · User interaction · Active learning

1 Introduction

With the advent and growing popularity of semantic web technologies in the last two decades, the number of applications relying on knowledge specified in terms of ontologies has considerably increased. One example of a vital field extensively adopting ontologies for highly critical applications is biomedicine.[1] As the size (up to hundreds of thousands of axioms) and complexity of the used ontologies is

This work is supported by Carinthian Science Fund (KWF), contract KWF-3520/26767/38701.

[1] See, e.g., OBO project (http://obo.sourceforge.net) or NCI-Thesaurus (http://ncit.nci.nih.gov).

C. Benzmüller et al. (Eds.): RuleML+RR 2018, LNCS 11092, pp. 164–184, 2018.
https://doi.org/10.1007/978-3-319-99906-7_11

constantly growing, the likeliness of faults, e.g. wrong entailments or logical contradictions, in these ontologies is significant. However, such defectiveness could have severe consequences, e.g. in health-related applications, on the one hand, and its root cause may be extremely hard to identify for humans on the other hand.

As a remedy, *ontology debugging (OD)* approaches [8,25], based on the general *model-based diagnosis* framework [15], have been developed. Given an ontology that does not satisfy requirements such as consistency, coherency or the (non-)entailment of certain axioms, the goal of OD is to find an explanation of the ontology's faultiness in terms of a set of incorrect axioms. Such an axiom set is called a *diagnosis*. However, the mere use of OD tools assisting a human by the generation (and ranking [9]) of diagnoses often does not solve the problem due to a couple of reasons. First [29], such (non-interactive) approaches often suggest unnecessarily large diagnoses (*non-parsimony*), neglect some solutions (*incompleteness*), return wrong explanations (*unsoundness*) or exhibit *poor performance*. Second, even if such a system overcomes all the said issues, it does unavoidably (due to a lack of additional information) suffer from the problem of generally large solution spaces (comprising up to thousands [27]) of competing diagnoses. Although the deletion or adequate modification of any diagnosis enables to formulate an ontology where all (initially present) faults are repaired, each diagnosis' choice leads necessarily to a solution ontology with different semantics [16]. Even if they are ranked, opting for (one of) the top-ranked diagnoses does not give any guarantees regarding the semantics of the resulting repaired ontology.

Addressing this issue, inspired by [3,11], (interactive) *test-driven ontology debugging (TOD)* techniques [16,24,27] were proposed and their general feasibility, scalability and practical efficiency was demonstrated by various conducted studies [21,22,27,28]. TOD techniques build on an idea well known from software engineering, which is the specification of test cases to successively narrow down the possible causes of fault. In the context of ontology engineering, a *positive (negative) test case* represents a set of axioms that must (not) be entailed by the intended ontology. The process of formulating test cases can be pursued until one diagnosis has overwhelming probability or, ultimately, until a single one remains. As the manual formulation of meaningful test cases – in the sense that they distinguish well between diagnoses – might be a hard task due to the involved mental reasoning with expressive logics (such as OWL 2 [6]), state-of-the-art TOD systems undertake the task of test case formulation and quality assessment. The workload for an *oracle*, usually a domain expert, interacting with the system thus reduces to classifying these automatically generated test cases, called *queries*, as positive or negative. This means answering questions whether presented axioms are or are not entailments of the intended ontology. Since oracle consultations are usually very costly, the practicality and efficiency of TOD approaches is inextricably linked to the number (and, e.g., the difficulty) of queries asked in order to pin down the *actual diagnosis*, i.e. the actually faulty axioms.

Unfortunately, the global minimization of the oracle costs (i.e. finding a cost-minimal *sequence of queries* revealing the actual diagnosis) is NP-hard [7]. As a result, TOD methods have to confine themselves to a local optimization (i.e. computing the best *next* query). To this end, a one-step lookahead evaluation of queries (i.e. how favorable is the expected situation after asking *one* query?) proved to be a very good trade-off between gained information and required effort [10], and is thus state-of-the-art in TOD.

However, there is not a unified view of what it means for a query to be "good", but several (one-step lookahead) heuristics [11,18,22,27], many of them inspired by active learning research [26]. These heuristics are expressed in terms of quantitative *query selection measures (QSMs)* that assign a real-valued goodness estimate to each query. A few empirical studies of QSMs in the domain of TOD exist, where [27,28] focus on two "traditional" QSMs [11,12] and [22] suggest and evaluate a novel QSM. Moreover, there are the theoretical analyses [17,18,21] which derive a range of new QSMs as well as equivalences and superiorities between new and traditional QSMs, and introduce efficient (heuristic) computation methods for optimal queries wrt. QSMs. Complementary to these researches, we shed light on the performance (in terms of oracle cost throughout a TOD session) of and relationships between *both the traditional and the new* QSMs in the present work. For this purpose, we are currently conducting extensive evaluations where we investigate the particular QSMs under varying conditions, similar to [27], regarding **(a)** diagnoses probability distributions, **(b)** quality (meaningfulness) of the probabilities, **(c)** available evidence (size of the diagnoses sample) for query computation, and **(d)** diagnostic structure (ontology size; # and size of diagnoses; reasoning complexity) using real-world debugging problems. The data of the (so-far finished) experiments shall be exploited to approach i.a. the following questions:

- Do the factors (a)–(d) affect the (relative) performance of the QSMs?
- Which QSM is preferable under which circumstances?
- Is there a (clear) winner among the QSMs?
- How far do the QSM performances differ under different conditions?

The rest of the paper is organized as follows. Section 2 briefly introduces technical basics wrt. TOD. Section 3 recaps the QSMs used in the experiments. The evaluation setting is described in Sect. 4, and results discussed in Sect. 5. Finally, Sect. 6 concludes.

2 Preliminaries

In this section we briefly characterize the basic technical concepts used throughout this work, based on the framework of [16,27] which is originally based on [15] (cf. [19]).

Ontology Debugging Problem Instance. An ontology to be debugged is given by $\mathcal{O} \cup \mathcal{B}$, where $\mathcal{O}$ includes the possibly faulty axioms and $\mathcal{B}$ the correct background knowledge axioms. That is, one can lay the debugging focus on just a subset $\mathcal{O}$ of the entire ontology, putting certain axioms, e.g. assertions, to $\mathcal{B}$. The pragmatics is that faults will only be sought within $\mathcal{O}$, i.e. the considered search space is restricted. Requirements to the intended ontology are captured by sets of positive (P) and negative (N) test cases [3]. 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) debugging problem instance (DPI).*

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

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

and assertions $\mathcal{A} : \{ax_4 : \mathit{ActiveResearcher}(ann)\}$. To debug the terminology while accepting as correct the assertion and stipulating that Ann is not necessarily an employee (negative test case $n_1 : \{\mathit{Employee}(ann)\}$), one can specify the following DPI: $dpi_{ex} := \langle \mathcal{T}, \mathcal{A}, \emptyset, \{n_1\} \rangle$. □

Diagnoses. Let $\mathbf{C}_\bot := \{C \sqsubseteq \bot \mid C \text{ class name in } \mathcal{O}\}$ and $U_P := \bigcup_{p \in P} p$. Given that the ontology to be debugged, 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 DPI $\langle \mathcal{O}, \mathcal{B}, P, N \rangle$ 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 suggest minimally-invasive repairs, modern TOD systems restrict the focus to the computation of minimal diagnoses.

Example 2. For $dpi_{ex} = \langle \mathcal{O}, \mathcal{B}, P, N \rangle$ 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 dpi_{ex}, as the deletion of any $ax_i \in \mathcal{O}$ breaks the unwanted entailment n_1. □

Diagnoses Probabilities. If axiom fault probabilities $p(ax_i)$ for $ax_i \in \mathcal{O}$ are available, e.g. by considering common fault patterns [14,23] or other heuristic information [9], probabilities of diagnoses $\mathcal{D} \in \mathbf{D}$ (of being the actual diagnosis) can be computed [11,27] as $p(\mathcal{D}) = \prod_{ax \in \mathcal{D}} p(ax) \prod_{ax \in \mathcal{O} \setminus \mathcal{D}} (1 - p(ax))$ and updated by means of Bayes' Rule (see [16, p. 130]) each time a new test case is added. Sometimes however $p(ax)$ for $ax \in \mathcal{O}$ might not be directly given, but derivable from the structure of the axioms. For instance, fault probabilities regarding logical (e.g. $\neg, \sqcap, \forall$) [27] or non-logical (e.g. class names) [16] symbols occurring in axioms might be available. Regarding the former, the axiom author might not properly use or fully understand these constructs from the logical perspective; as to the latter, the axiom author, say an orthopedist, might not possess the required (domain) expertise regarding certain concepts, say *Acne* or *Basalioma*. Such fault information may originate from, e.g., experience, a subjective or expert guess, or through the analysis of relevant logs or past debugging sessions. Let $p(s_i)$ be the probability that a (non-)logical symbol s_i is faulty and n_i be the number of occurrences of s_i in ax. Then [27]: $p(ax) = 1 - \prod_{s_i \text{ occurs in } ax}(1 - p(s_i))^{n_i}$.

Example 3. Reconsider dpi_{ex} from Example 1. Suppose that the ontology author knows from past debugging sessions to make quite many mistakes using quantifiers, some using negation, conjunction and disjunction, but almost none using subsumption. This could lead to the fault probabilities $\langle p(\exists), p(\sqcap), p(\sqcup), p(\sqsubseteq)\rangle = \langle 0.25, 0.05, 0.05, 0.01\rangle$ relevant to dpi_{ex}. Using these, the fault probability of axiom ax_1 (including the symbols $\sqsubseteq, \exists, \sqcup$) computes as $p(ax_1) = 1 - (1 - 0.01)(1-0.25)(1-0.05) \approx 0.29$. Similarly, we obtain $p(ax_2) \approx 0.26$ and $p(ax_3) \approx 0.06$. Hence, we can derive $p(\mathcal{D}_1) = (0.29)(1 - 0.26)(1 - 0.06) \approx 0.21$, $p(\mathcal{D}_2) \approx 0.17$ and $p(\mathcal{D}_3) \approx 0.03$. □

Queries and Q-Partition. Let $\mathbf{D}$, called the *leading diagnoses*, be a set of at least two (precomputed) minimal diagnoses for $dpi = \langle \mathcal{O}, \mathcal{B}, P, N\rangle$. Usually, the diagnoses with highest probability or minimum cardinality are used for this purpose. A *query* (wrt. $\mathbf{D}$) is a set of axioms q that rules out at least one diagnosis in $\mathbf{D}$, both if q is classified as a positive test case ($P \leftarrow P \cup \{q\}$), and if q is classified as a negative test case ($N \leftarrow N \cup \{q\}$). That is, 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 diagnosis $\mathcal{D}_j \in \mathbf{D}$ is not a diagnosis for $\langle \mathcal{O}, \mathcal{B}, P, N \cup \{q\}\rangle$. The classification of a query q to either P or N is accomplished by an oracle, e.g. a domain expert, answering the question "Is (each axiom in) q an entailment of the intended (correct) ontology?". Thus, the *oracle* is a function $\mathsf{class} : \mathbf{Q} \to \{P, N\}$ where $\mathbf{Q}$ is the query space; $\mathsf{class}(q) = P$ if the answer to the question is positive, else $\mathsf{class}(q) = N$.

An expedient tool towards the verification and goodness estimation of query candidates q is the notion of a q-partition. Namely, every set of axioms q partitions any set of diagnoses $\mathbf{D}$ into three subsets:

- $\mathbf{D}_q^+$: includes all $\mathcal{D} \in \mathbf{D}$ where $\mathcal{D}$ is not a diagnosis for $\langle \mathcal{O}, \mathcal{B}, P, N \cup \{q\}\rangle$ (diagnoses predicting that q is a positive test case)
- $\mathbf{D}_q^-$: includes all $\mathcal{D} \in \mathbf{D}$ where $\mathcal{D}$ is not a diagnosis for $\langle \mathcal{O}, \mathcal{B}, P \cup \{q\}, N\rangle$ (diagnoses predicting that q is a negative test case)
- $\mathbf{D}_q^0 = \mathbf{D} \setminus (\mathbf{D}_q^+ \cup \mathbf{D}_q^-)$: includes all $\mathcal{D} \in \mathbf{D}$ where $\mathcal{D}$ is a diagnosis for both $\langle \mathcal{O}, \mathcal{B}, P \cup \{q\}, N\rangle$ and $\langle \mathcal{O}, \mathcal{B}, P, N \cup \{q\}\rangle$
(*uncommitted diagnoses*, no prediction about q)

A partition $\mathfrak{P}$ of $\mathbf{D}$ into three sets is called *q-partition* iff there is a query q wrt. $\mathbf{D}$ such that $\mathfrak{P} = \langle \mathbf{D}_q^+, \mathbf{D}_q^-, \mathbf{D}_q^0 \rangle$. According to the definition of a query, it holds that q is a query iff both $\mathbf{D}_q^+$ and $\mathbf{D}_q^-$ are non-empty sets. This fact can be taken advantage of for *query verification*. Coupled with diagnoses probabilities, the q-partition provides useful hints [18] about the *query quality* in that it enables to

(1) test whether q is a *strong query*, i.e. one without uncommitted diagnoses ($\mathbf{D}_q^0 = \emptyset$),
(2) estimate the impact q's classification $\mathsf{class}(q)$ has in terms of diagnoses elimination (potential a-posteriori change of the diagnoses space), and
(3) assess the probability of q's positive and negative classification (e.g. to compute the uncertainty of q).

For given $\mathbf{D}$, we estimate [11]: $p(\mathsf{class}(q) = P) = p(\mathbf{D}_q^+) + \frac{1}{2}p(\mathbf{D}_q^0)$ and $p(\mathsf{class}(q) = N) = p(\mathbf{D}_q^-) + \frac{1}{2}p(\mathbf{D}_q^0)$ where $p(\mathbf{D}_q^X) = \sum_{\mathcal{D} \in \mathbf{D}_q^X} p(\mathcal{D})$ for $X \in \{+,-,0\}$ and $p(\mathcal{D})$ for $\mathcal{D} \in \mathbf{D}$ is the probability of $\mathcal{D}$ normalized over $\mathbf{D}$ (i.e. $\sum_{\mathcal{D} \in \mathbf{D}} p(\mathcal{D}) = 1$).

Example 4. Let the computed leading diagnoses for dpi_{ex} be $\mathbf{D} = \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$. One query wrt. $\mathbf{D}$ is, e.g., $q_1 := \{\mathit{ActiveResearcher} \sqsubseteq \mathit{Author}\}$. Because, (a) 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, (b) 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. The resulting q-partition for q_1 is thus $\langle \mathbf{D}_{q_1}^+, \mathbf{D}_{q_1}^-, \mathbf{D}_{q_1}^0 \rangle = \langle \{\mathcal{D}_3\}, \{\mathcal{D}_1, \mathcal{D}_2\}, \emptyset \rangle$. Consequently, q_1 is a strong query ($\mathbf{D}_{q_1}^0 = \emptyset$) and the estimated probability of q_1's positive (negative) classification, based on the normalized diagnoses probabilities $\langle p(\mathcal{D}_1), \dots, p(\mathcal{D}_3) \rangle = \langle 0.5, 0.42, 0.08 \rangle$, is 0.08 (0.92). Note, e.g., $q_2 := \{\mathit{Author} \sqsubseteq \mathit{Person}\}$, having the partition $\langle \{\mathcal{D}_1, \mathcal{D}_2\}, \emptyset, \{\mathcal{D}_3\} \rangle$, is not a query since no leading diagnoses are invalidated after assigning q_2 to P, i.e. a positive answer does not bring along any useful information for diagnoses discrimination. Intuitively, 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). □

Test-Driven Ontology Debugging. Formally, the (optimal) test-driven ontology debugging problem (TOD) can be stated as follows:

Problem 1 ((Optimal) TOD). **Given:** DPI $\langle \mathcal{O}, \mathcal{B}, P, N \rangle$. **Find:** (Lowest-cost) set of test cases $P' \cup N'$ such that there is only one minimal diagnosis for $\langle \mathcal{O}, \mathcal{B}, P \cup P', N \cup N' \rangle$.

Example 5. Let the actual diagnosis 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 dpi_{ex} as an input, solutions to the TOD problem, 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 TOD cost by the number of test cases, the latter solution (cost: 1) is optimal, the former (cost: 2) not. □

Note, TOD is a *symptom-driven* approach to *fault localization*. That is, given some discrepancies (symptom), such as inconsistency or unwanted entailments, between the actual and the intended ontology, the goal is to (efficiently) collect sufficient information to locate the faulty axioms (actual diagnosis) *that explain the observed problems*. TOD must be distinguished from, but can nevertheless be profitably combined with, other techniques, e.g. ones addressing *ontology repair* [8,30] (how to correct the faulty axioms?), *ontology revision* [13] (find *all* faulty axioms) or *ontology enrichment* [5] (find missing axioms; can help to detect problems/symptoms).

Query Selection Measures (QSMs). The said query properties (1)–(3) characterized by the q-partition are essentially what QSMs take into account to quantitatively rate the query quality. Formally, a QSM is a function $m : \mathbf{Q} \to \mathbb{R}$ that assigns a value $m(q)$ to each query $q \in \mathbf{Q}$. All QSMs are heuristics towards Optimal TOD (Problem 1). That is, their goal is to minimize the expected cost $\sum_{\mathcal{D}} p(\mathcal{D})\mathsf{cost}(\mathcal{D})$ of locating the actual diagnosis $\mathcal{D}^*$. At this, $\mathsf{cost}(\mathcal{D})$ is usually conceived of as the sum of individual query (answering) costs over all queries required to unambiguously isolate $\mathcal{D}$. For the purpose of this paper we assume $\mathsf{cost}(\mathcal{D})$ represents the number of queries to isolate $\mathcal{D}$ (all queries assumed equally costly).

Table 1. ([18, Table 2]) QSM designators (column 1) and according functions $m(q)$ (column 2). Column 3 indicates whether the QSM is optimized by maximizing (↗) or minimizing (↘) the function m.

QSM m	$m(q)$	opt.
ENT	$\sum_{c\in\{P,N\}} p(\mathsf{class}(q)=c)\log_2 p(\mathsf{class}(q)=c)$	↘
SPL	$\left\lvert \lvert\mathbf{D}_q^+\rvert - \lvert\mathbf{D}_q^-\rvert \right\rvert$	↘
KL	$-\sum_{X\in\left\{\mathbf{D}_q^+,\mathbf{D}_q^-\right\}} \frac{\lvert X\rvert}{\lvert\mathbf{D}_q^+\cup\mathbf{D}_q^-\rvert}\log_2\frac{p(X)}{p(\mathbf{D}_q^+\cup\mathbf{D}_q^-)}$	↗
EMCb	$p(\mathsf{class}(q)=P)\lvert\mathbf{D}_q^-\rvert + p(\mathsf{class}(q)=N)\lvert\mathbf{D}_q^+\rvert$	↗
MPS	$p(\mathbf{D}_{q,\min})$ if $\lvert\mathbf{D}_{q,\min}\rvert = 1$, 0 else **1)**	↗
BME	$\lvert\mathbf{D}_{q,p,\min}\rvert$ **2)**	↗
RIO′	$\frac{\mathsf{ENT}(Q)}{2} + \mathbf{D}_{q,n}$ **3)**	↘

Key:

1): $\mathbf{D}_{q,\min} := \arg\min_{X\in\left\{\mathbf{D}_q^+,\mathbf{D}_q^-\right\}}(|X|)$

2): $\mathbf{D}_{q,p,\min} :=$
$$\begin{cases}\mathbf{D}_q^- & \text{if } p(\mathbf{D}_q^-) < p(\mathbf{D}_q^+)\\ \mathbf{D}_q^+ & \text{if } p(\mathbf{D}_q^+) < p(\mathbf{D}_q^-)\\ 0 & \text{else}\end{cases}$$

3): $\mathbf{D}_{q,n} := \begin{cases} c_q - n & \text{if } c_q \geq n\\ |\mathbf{D}| & \text{else}\end{cases}$

where $c_q := \min\{|\mathbf{D}_q^+|, |\mathbf{D}_q^-|\}$ and n denotes the minimal number of diagnoses the selected query must eliminate [22]

3 The Evaluated Heuristics

In this section we briefly revisit and explain the QSMs – originally introduced in other works – we use in our experiments. These include the "classical" frequently used ones [11,27] and the newer ones proposed in [18,22] and discussed in-depth in [17]. Since we employ a query computation and selection method [21] that guarantees to produce only (the more favorable, cf. [20, Sect. 2.4.1]) strong queries, [18, Table 3] tells us that we have to deal with seven (non-equivalent) QSMs in this case. We next illustrate the rough idea behind these heuristics, listed in Table 1. Note, we also mention a random QSM which we used as a baseline in our evaluations.

Information Gain ENT**:** [11,27] Chooses a query with the highest expected information gain or, equivalently, with the lowest expected posterior entropy wrt. the diagnoses set $\mathbf{D}$. As derived in [11], $\mathsf{ENT}(q)$ is the better, the closer the probabilities for positive and negative classification of q are to 0.5 (cf. formula in Table 1).

Split-In-Half SPL**:** [12,27] Chooses a query q whose q-partition best splits the diagnoses set $\mathbf{D}$ in half, i.e. where both $|\mathbf{D}_q^+|$ and $|\mathbf{D}_q^-|$ are closest to $\frac{1}{2}|\mathbf{D}|$. Intuitively, an optimal q wrt. SPL guarantees that a half of the (known) diagnoses are eliminated by querying q's classification.

Kullback-Leibler Divergence KL**:** [17,18,26] Chooses a query with largest average disagreement between query-classification predictions of single diagnoses $\mathcal{D} \in \mathbf{D}$ and the consensus (prediction) of all $\mathcal{D} \in \mathbf{D}$, based on an information-theoretic measure of the difference between two probability distributions [26]. As demonstrated in [17, Prop. 26], this QSM can be represented in terms of the formula given in Table 1.

Expected Model Change EMCb**:** [17,18,26] Chooses a query for which the expected number of invalidated diagnoses in $\mathbf{D}$ is maximized.

Most Probable Singleton MPS**:** [17,18] Chooses a query q for which the minimum-cardinality set among $\left\{\mathbf{D}_q^+, \mathbf{D}_q^-\right\}$ is a singleton $\{\mathcal{D}\}$ where $\mathcal{D}$ has maximal probability. Intuitively, MPS seeks to eliminate, with a maximal probability, the maximal possible number of $|\mathbf{D}| - 1$ diagnoses in $\mathbf{D}$.

Biased Maximal Elimination BME**:** [17,18] Chooses a query with a bias (probability >0.5) towards one classification (P or N) such that this more likely classification rules out an as high as possible number of diagnoses in $\mathbf{D}$.

Risk Optimization RIO'**:** [17,22] Chooses a query with optimal information gain (ENT-value) among those that, in the worst case, eliminate (at least) $n \leq \frac{1}{2}|\mathbf{D}|$ diagnoses in $\mathbf{D}$. At this, the parameter n is learned by reinforcement based on the diagnoses elimination performance achieved so far during a TOD session.[2]

[2] We consider the slightly modified version RIO' of the original RIO [22], as suggested in [18].

Table 2. q-partitions for the queries $q_i := \{ax_i\}$ in Example 6.

q	$\mathbf{D}^+_q$	$\mathbf{D}^-_q$	$p(\mathbf{D}^+_q)$	$p(\mathbf{D}^-_q)$
$\{ax_1\}$	$\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_6\}$	$\{\mathcal{D}_5\}$	0.59	0.41
$\{ax_2\}$	$\{\mathcal{D}_5, \mathcal{D}_6\}$	$\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4\}$	0.45	0.55
$\{ax_3\}$	$\{\mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_5\}$	$\{\mathcal{D}_1, \mathcal{D}_6\}$	0.95	0.05
$\{ax_4\}$	$\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4\}$	$\{\mathcal{D}_5, \mathcal{D}_6\}$	0.55	0.45
$\{ax_5\}$	$\{\mathcal{D}_1, \mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_5, \mathcal{D}_6\}$	$\{\mathcal{D}_2\}$	0.67	0.33
$\{ax_6\}$	$\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_4, \mathcal{D}_5, \mathcal{D}_6\}$	$\{\mathcal{D}_3\}$	0.86	0.14
$\{ax_7\}$	$\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$	$\{\mathcal{D}_4, \mathcal{D}_5, \mathcal{D}_6\}$	0.48	0.52

Random RND**:** Samples one element uniformly at random from the query space $\mathbf{Q}$.

Example 6. To illustrate these different selection principles, let us consider a DPI (cf. [17, Table 1 + 2]) with $\mathcal{O} = \{1, \dots, 7\}$ (where numbers i denote axioms ax_i) which gives rise to the minimal diagnoses $\mathbf{D}$ and associated diagnoses probabilities given by

$$\mathbf{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_6\} = \{[2,3], [2,5], [2,6], [2,7], [1,4,7], [3,4,7]\}$$
$$\{p(\mathcal{D}_1), \dots, p(\mathcal{D}_6)\} = \{0.01, 0.33, 0.14, 0.07, 0.41, 0.04\}$$

Let, for simplicity, the query space $\mathbf{Q}$ consist (only) of all single axiom sets $q_i := \{ax_i\}$ for $ax_i \in \mathcal{O}$. The q-partitions of these queries are shown in Table 2. Now, the query choice made by the discussed QSMs in this case is as given in Table 3. □

Table 3. Query choice made by the different QSMs in Example 6.

QSM	best query	explanation
ENT:	q_7	$p(\mathbf{D}^+_{q_7})$ and $p(\mathbf{D}^-_{q_7})$ are closest to 0.5 over all queries q_i
SPL:	q_7	$\lvert\mathbf{D}^+_{q_7}\rvert$ and $\lvert\mathbf{D}^-_{q_7}\rvert$ are equal to $\frac{\lvert\mathbf{D}\rvert}{2} = 3$
KL:	q_3	$\mathsf{KL}(q_3) = 1.48$ is maximal over all queries q_i
EMCb:	q_7	the expected number of eliminated diagnoses is 3 (and lower for all other queries q_i)
MPS:	q_1	$\lvert\mathbf{D}^-_{q_1}\rvert = \lvert\{\mathcal{D}_5\}\rvert = 1$ and $p(\mathcal{D}_5) = 0.41 > 0.33$(cf. q_5) > 0.14(cf. q_6)
BME:	q_7	$\mathsf{BME}(q_7) = 3$ is maximal over all queries q_i
RIO$'$ (with $n = 2$):	q_2 or q_4	these are the queries with best ENT-value among all queries ($q2$, $q3$, $q4$) which eliminate n diagnoses in $\mathbf{D}$ in the worst case

4 Experimental Settings

The Dataset. Table 4 depicts the (part of the overall) dataset investigated in the so-far[3] finished experiments. The tested ontologies U,M,T,E are inconsistent real-world cases; these were also examined in [8,27]. The DPI dpi_j we extracted from $\mathcal{O}_j$ was $\langle \mathcal{O}_j, \emptyset, \emptyset, \emptyset \rangle$ ($j = 1, \dots, 4$), i.e. the background $\mathcal{B}$, positive (P) and negative (N) test cases were (initially) empty. Moreover, Table 4 shows the diagnostic structure of the used debugging problems in terms of the ontology size, the number and size of minimal diagnoses, and the logical expressivity which influences the reasoning complexity.

The Factors. To test the behavior and robustness of the discussed QSMs under various scenarios, we – in addition to the DPI – varied the following factors in our experiments:

(F1) the type of probability distribution (concerning faults wrt. logical symbols) (*non-biased*, *moderately biased*, *strongly biased*),
(F2) 3 different random choices of assigned probabilities for each distribution type (to average out potential peculiarities of a specific probability assignment),
(F3) the plausibility of the probabilities (simulated by *plausible*, *random*, *implausible* oracle behavior),
(F4) the amount of information available for query selection (number of leading diagnoses $ld \in \{6, 10, 14\}$), and
(F5) the actual diagnosis $\mathcal{D}^*$ (i.e. the target solution of the TOD sessions).

Ad (F1): Let S be the set of all logical symbols (cf. Sect. 2) occurring over all $ax \in \mathcal{O}_j$ and $E_\lambda(x) = \lambda e^{-\lambda x}$ the probability density function of the exponential distribution. Three probability distribution types were modeled, by assigning to each symbol in $S \dots$

- *all-equal (EQ)*: ...a fixed equal (random) value $r \in [0, 1]$
- *moderately biased (MOD)*: ...the probability $E_\lambda(x_i)$ for a random $x_i \in [i - \frac{1}{2}, i + \frac{1}{2})$ where i is randomly chosen (without replacement) from $\{1, \dots, |S|\}$ and $\lambda := 0.5$ (same λ as used in [27])
- *strongly biased (STR)*: ...the probability $E_\lambda(x_i)$ for a random $x_i \in [i - \frac{1}{2}, i + \frac{1}{2})$ where i is randomly chosen (without replacement) from $\{1, \dots, |S|\}$ and $\lambda := 1.75$ (same λ as used in [27])

[3] Note, due to the comprehensiveness (large number of factor combinations tested) of our evaluations, experiments are very time-consuming (up to several weeks for one ontology).

Table 4. Dataset used in the experiments.

j	KB $\mathcal{O}_j$	$\lvert\mathcal{O}_j\rvert$	expressivity 1)	#D/min/max 2)
1	University (U) 3)	50	$\mathcal{SOIN}^{(D)}$	90/3/4
2	MiniTambis (M) 3)	173	$\mathcal{ALCN}$	48/3/3
3	Transportation (T) 3)	1300	$\mathcal{ALCH}^{(D)}$	1782/6/9
4	Economy (E) 3)	1781	$\mathcal{ALCH}^{(D)}$	864/4/8

Key:

1): Description Logic expressivity [1, p. 525ff.].

2): #D, min, max denote the number, the min. and max. size of minimal diagnoses for the DPI.

3): Sufficiently complex cases (#D $\geq$ 40) used in [27].

Intuitively, one can view both MOD and STR to (1) precompute a sequence $p_1 > \dots > p_{|S|}$ of values in $(0, 1)$ where, on average, the ratio between each value p_i and the next smaller one p_{i+1} is $p_i/p_{i+1} = e^\lambda$, i.e. ≈ 1.6 for MOD and ≈ 5.8 for STR, and (2) assign to each $s \in S$ a randomly chosen probability p_i from this sequence without replacement. Hence, if sorted from large to small, the fault probabilities assigned to logical symbols occurring in $\mathcal{O}_j$ are completely uniform for EQ (*no bias*), moderately descending for MOD (*moderate bias*) and steeply descending for STR (*strong bias*).

Example 7. Returning to Example 3, note the strong bias (STR) in the fault probabilities of the symbols $\exists, \sqcap, \sqsubseteq$, i.e. each probability is five times as high as the next one. □

For instance, EQ could model a situation where a novice knowledge engineer or domain expert obtains a faulty ontology, and there is no relevant information about their faults at hand. On the other hand, MOD can be interpreted to simulate a moderate tendency in the fault information, i.e. a non-negligible number of symbols have a non-negligible fault probability. For example, an ontology author might extract from logs of her past debugging sessions that she were misusing a range of different symbols, but some more often than others. STR reflects cases where the differences in fault likeliness are substantial, i.e. very few symbols have a non-negligible probability of being faulty, whereas most of the symbols are practically always correct. An example is a knowledge engineer that, in the past, has made almost only errors regarding quantifiers.

Ad (F3): Let q be a query with $p(\mathsf{class}(q) = P) = x$ (cf. Sect. 2). The plausibility of the given probabilistic information was modeled by different oracle functions class, simulating different strategies of query classification:

- *plausible*: classify q to P with probability x
- *random*: classify q to P with probability 0.5
- *implausible*: classify q to P with probability $1 - x$

Recall (Sect. 2), $p(\mathsf{class}(q) = N) = 1 - p(\mathsf{class}(q) = P)$. The idea is that, given (un)reasonable fault information, the estimated query classification probabilities should be good (bad) approximations of the real likeliness of getting a respective outcome. The plausible scenario reflects the case where given probabilities are useful and provide a rational bias, e.g. when different (reliable) sources of fault information are integrated or the user knows their strengths and weaknesses well. The random strategy aims at estimating the average number of queries needed to pin down $\mathcal{D}^*$, assuming we cannot make useful predictions about the oracle. The implausible strategy represents a misleading fault model, where probabilities turn out to be opposite to what the given information suggests, e.g., when using subjective estimates or historical data that do not apply to the present scenario. As QSMs utilize fault information for query suggestion, we want to assess their robustness under changing fault information quality [22,27].

Example 8. Intuitively, given the actual diagnosis $\mathcal{D}_3$ in our running example (Examples 1–5) and the diagnoses probability distribution in Example 3 (assigning $\mathcal{D}_3$ only a value of 0.03), the fault information $p(.)$ would fall into the category "implausible". □

Ad (F5): We specified the target solution $\mathcal{D}^*$ in the different TOD sessions implicitly through the oracle answer strategies, see (F3). That is, each TOD session continued until positive (P') and negative (N') test cases were collected such that there was just a single minimal diagnosis for the DPI $\langle \mathcal{O}_j, \emptyset, P', N' \rangle$ (resulting from the initial DPI by adding P' and N', cf. Problem 1). This implicit definition of $\mathcal{D}^*$ has the advantage of higher generality and closeness to reality over a prior explicit fixation of some $\mathcal{D}^*$ among the minimal diagnoses for the initial DPI $dpi_j^0 := \langle \mathcal{O}_j, \emptyset, \emptyset, \emptyset \rangle$ [27]. Because, in the latter case only one specific class of TOD problems is considered, namely those where the actual solution $\mathcal{D}^*$ is already a minimal diagnosis for dpi_j^0. In practice, this assumption might often not hold. The reason is that the DPI changes throughout TOD, i.e. dpi_j^i becomes dpi_j^{i+1} after the incorporation of a new test case; this transformation generally gives rise to "new" diagnoses (minimal diagnoses for dpi_j^{i+1}) that are proper supersets of ruled out"original" ones (minimal diagnoses for dpi_j^i inconsistent with the added test case) [15,16].

The Tests. For each of the DPIs $dpi_1, \dots, dpi_4$, for each of the 8 QSMs explicated in Sect. 3, and for each of the 3^4 factor level combinations of factors (F1) – (F4) we performed 20 TOD sessions, adopting the algorithms for query computation presented in [20,21]. Factor (F5) was implicitly varied in these 20 runs through the randomized oracle behavior (F3), yielding in most cases a different $\mathcal{D}^*$. When some $\mathcal{D}^*$ happened to occur repeatedly in the 20 sessions, we discarded such duplicate runs.[4]

[4] To reproduce the experiments or access logs see http://isbi.aau.at/ontodebug/evaluation.

5 Experimental Results

5.1 Representation

The obtained experimental results are shown by Figs. 1, 2, 3 and 4 which graph the number of queries required by the tested QSMs until $\mathcal{D}^*$ could be isolated. At this, the green/yellow/red bars depict the situation of a plausibly/randomly/implausibly answering oracle (F3). Each bar represents an average over (up to) 20 TOD sessions (F5) and 3 random choices of probabilities (F2). Each figure summarizes the results for one ontology in Table 4; the plots for the U and T cases are more comprehensive, including all combinations of factor levels for (F1), (F3) and (F4), whereas the depictions of M and E are kept shorter due to space restrictions, showing only the $ld = 10$ case of (F4) for all settings of (F1) and (F3). Along the x-axes of the figures we have the 8 different QSMs, grouped by manifestations of factor (F4) in Figs. 1 and 3, and by instantiations of factor (F1) in Figs. 2 and 4.

5.2 Observations

Gained insights from the study of the experimental data are discussed next.

Is there a Clear Winner? This question can be answered negatively pretty clearly. For instance, have a look at the MOD, $ld = 14$ case in Fig. 1. Here we see that MPS performs really good compared to all other QSMs for all oracle types. In fact, it is better than all others in the plausible and random configurations, and loses just narrowly against RND given implausible answers. However, if we draw our attention to, e.g., the EQ case in the same figure, we recognize that MPS comes off significantly worse than other heuristics under a plausible oracle behavior. Similar argumentations apply for all other potential winner QSMs. For $ld = 10$, Table 5, which lists the best QSMs in all the different settings we investigated, confirms that there is no single best QSM.

Sensitivity to Fault Information. That there is no QSM which always outmatches all others is not a great surprise, as we evaluate under various types of given probabilistic information $p(.)$ and the different measures exploit $p(.)$ to a different extent when selecting a query. As a result, we can observe probability-independent QSMs such as SPL outperform (lose against) strongly probability-reliant ones such as ENT in situations where the fault information is wrongly (reasonably) biased, e.g., see the implausible (plausible) cases for MOD and STR in Figs. 1 and 3. So, e.g., SPL can never benefit from high-quality meta information about faults, but cannot effect a significant overhead given low-quality probabilities either. The behavior of, e.g., ENT, is diametrically opposite. To verify this, check the difference between the green and red bars for both SPL and ENT for MOD and STR; for SPL they are hardly different at all, whereas for ENT they diverge rapidly as we raise the bias (EQ → MOD → STR) in the underlying distribution. In contrast to these extreme cases, there is, e.g., RIO' which incorporates both the diagnoses elimination rate and fault probabilities in its calculations. The consequence is a behavior that mostly lies in between the performances of SPL and ENT. Based on the data in the figures, which is quite consistent in this regard, the following *qualitative ordering from most to least probability-sensitive* can be imposed on QSMs:

$$\langle \mathsf{EMCb}, \mathsf{BME}, \mathsf{ENT}, \mathsf{KL}, \mathsf{MPS}, \mathsf{RIO}', \mathsf{RND}, \mathsf{SPL} \rangle \tag{1}$$

Impact of the DPI/Diagnostic Structure. Trivially, the overall number of (minimal) diagnoses to discriminate between impacts the average number of queries required. Thus, for M (48 minimal diagnoses initially), U (90), E (864) and T (1782), respectively, the min/avg/max number of queries over all QSMs and sessions is (rounded) 3/7/18, 4/8/19, 6/10/19 and 4/12/29. The difference between M and E, for instance, can be quite well seen by comparing the length of the bars in Figs. 2 and 4 which are placed side by side. On the contrary and as one would expect, there are no indications of the ontology size $|\mathcal{O}_j|$ (3rd column, Table 4) having a remarkable influence on QSM performance (as the ontology size has generally no bearing on the number of minimal diagnoses). The reasoning complexity (4th column, Table 4), in contrast, albeit not relevant to the QSM performance, is known to affect the query computation time [20]. The latter was quite constant over all runs and QSMs and amounted to maximally 0.18/0.14/0.18/0.13 sec (per query) for the cases M/U/E/T. The relative behavior of the QSMs under varying DPI (but otherwise same conditions) appears to be quite stable. To see this, compare, e.g., the EQ, the MOD and the STR cases between Figs. 1 and 3, or Figs. 2 and 4. From the pragmatic point of view, if this consistency of QSM performances irrespective of the particular DPI generalizes (as needs to be verified using a larger dataset), a nice implication thereof would be the possibility to recommend (against) QSMs independently of (the structure of) the problem at hand.

Impact of the Leading Diagnoses. As Figs. 1 and 3 indicate quite well, and numbers confirm, there is no significant average difference in the numbers of queries for varying $ld \in \{6, 10, 14\}$. This is in line with the findings of [2]. What we can realize, though, is an exacerbation of the discrepancy between the plausible (green bars) and implausible (red bars) cases when ld increases. The random case (yellow bars), on the other hand, is mostly stable. The reason for this intensification of the effect of good or bad bias with larger diagnoses samples is that more extreme decisions might be made in this case. A simple illustration of this is to compare a "risky" [22] query (one that might invalidate very few diagnoses) wrt. a sample of 3 and 100 diagnoses; in the former case, this would be one eliminating either 1 or 2, in the latter one ruling out either 1 or 99 known hypotheses. We see that the former query is similar to a"risk-less" split-in-half choice, while the latter is far off being that conservative. A practical consequence of this is that it might make sense to try generating a higher number of diagnoses per iteration (if feasible in reasonable time) if a probability-based measure, e.g. EMCb or ENT, is used and the trust in the given (biased) fault information is high (e.g. if reliable historical data is available). Verify this by considering

Table 5. Shows which QSM(s) exhibited best performance in the various scenarios in (F1) × (F3) for all DPIs (1st column) in Table 4 and the setting $ld = 10$ of (F4). The QSM(s) with lowest # of queries (per scenario) are underlined. All stated non-underlined QSMs lay within 3% of the best QSM wrt. # of queries. The number below the QSM(s) gives the possible overhead $(\#q_{worstQSM(S),S}/\#q_{bestQSM(S),S} - 1) * 100$ in % incurred by using a non-optimal QSM in a scenario S, where $\#q_{X,S}$ refers to the # of required queries of QSM X in scenario S, and $bestQSM(S)$ / $worstQSM(S)$ denote the best/worst QSM in scenario S. The colors indicate criticality of QSM choice based on the overhead, from lowest = green to highest = red.

	PLAUSIBLE			RANDOM			IMPLAUSIBLE		
	EQ	MOD	STR	EQ	MOD	STR	EQ	MOD	STR
M	KL	RIO′, ENT, BME	MPS, ENT	MPS	MPS	MPS	MPS	RND	RND
	63	176	144	46	47	48	118	131	277
U	BME	BME	MPS, BME	MPS	MPS	MPS	RND	RND	RND, KL
	59	129	151	42	50	53	67	149	220
E	BME	ENT	EMCb, RIO′, BME	ENT, MPS	MPS	ENT, RIO′, BME, MPS	MPS	KL	RND
	64	93	90	30	33	37	121	191	93
T	EMCb	EMCb, RIO′, ENT	ENT, BME, EMCb, MPS	MPS	MPS	MPS	MPS	RND	RND
	62	125	174	45	40	38	93	102	123

Table 6. Number of times each QSM is (among) the best in Table 5.

		ENT	SPL	KL	EMCb	MPS	BME	RIO′	RND
ALL	among best	7	0	3	4	18	8	4	8
	the best	4	0	2	4	16	4	2	8
PLAUSIBLE	among best	5	0	1	4	3	7	3	0
	the best	2	0	1	4	3	4	1	0

EMCb and ENT in the MOD and STR cases for $ld \in \{6, 14\}$ in Figs. 1 and 3. By contrast, when adopting a probability-insensitive QSM, say SPL, one seems to be mostly better off when relying on a smaller ld. That is, when the meta information is vague, a good option is to rely on a "cautious" [22] measure such as SPL *and* a small diagnoses sample. Note, the latter is doubly beneficial as it also decreases computation times.

Importance of Using a Suitable QSM. To quantify the importance of QSM choice we compute the *degree of criticality of choosing the right QSM in a scenario* as the overhead in oracle cost (number of queries) when employing the worst instead of the best QSM in this scenario, see (the caption of) Table 5. At this, a *scenario* refers to one factor level combination in (F1) × (F3). We learn from Table 5 that, even in the least critical cases (green-colored), we might experience a worst-case overhead in oracle effort of at least 30% when opting for the wrong QSM. This overhead is drastically higher in other cases and reaches figures of over 250%. That is, more than triple the effort might be necessary to locate a fault under an inopportune choice of QSM heuristic. However, we emphasize that even a 30% overhead must be considered serious given that usually oracle inquiries are very costly. Hence, appropriate QSM selection *is* an important issue to be addressed in all scenarios.

As a predictor of the criticality, the scenario (columns in Table 5) appears to be a reasonable candidate, as the colors already suggest. In fact, the coefficients of variation, one computed for each column in Table 5, are fairly low, ranging from 3% to 26% (except for the last column with 47%). So, the negative effect of a bad QSM choice is similar in equal scenarios, and does not seem to be dependent on the DPI.

Which QSM to use in which Scenario? To approach this question, we have, for all four DPIs, analyzed all the nine settings in (F1) × (F3) wrt. the optimal choice of a QSM. The result is presented in Table 5. We now discuss various insights from this analysis.

Overall Picture. SPL is never a (nearly) optimal option. This is quite natural because, intuitively, going for no "risk" at all means at the same time excluding the chance to perform extraordinarily well. All other QSMs appear multiple times among those QSMs which are $\leq$3% off the observed optimal number of queries. Table 6 (rows 1 + 2) lists how often each QSM is (among) the best. It shows that MPS is close to the optimum in a half of the cases, significantly more often than all other heuristics. However, blindly deciding for MPS is not a rational way to go. Instead, one must consider the numbers at a more fine-grained level, distinguishing between the quality of the given fault distribution (blocks in Table 5), to get a clearer and more informative picture.

The Implausible Cases: Here RND distinctly prevails. It occurs in all but four optimal QSM sets, and is often *much* better than other measures, e.g., see the STR setting in Fig. 2. At first sight, it might appear counterintuitive that a random selection outweighs all others. One explanation is simply that the randomness prevents RND from getting misled by the (wrong) fault information. Remarkable is, however, that in quasi all cases RND significantly outperforms SPL, which acts independently of the given probabilities as well. The conclusion from this is that, whenever the prior distribution is wrongly biased, introducing randomness into the query selection procedure saves oracle effort.

The Random Cases: These cases are strongly dominated by MPS which occurs in each set of best QSMs per scenario. Therefore, whenever the given fault information does neither manifest a tendency towards nor against the actual diagnosis, MPS is the proper heuristic. Moreover, the benefit of using MPS seems to increase the more leading diagnoses are available for query selection (see Figs. 1 and 3). Since MPS, in attempt to invalidate *a maximal number of diagnoses*, suggests very "risky" queries (see above), a possible explanation for this is that acting on a larger diagnoses sample allows to guarantee a higher risk than when relying on a smaller sample (cf. discussion above). However, as all Figs. 1, 2, 3 and 4 clearly reveal, MPS is definitely the wrong choice in any situation where we have a plausible, but unbiased probability distribution. In such cases it manifests sometimes significantly worse results than other heuristics do. But, as soon as a bias is given, the performance of MPS gets really good.

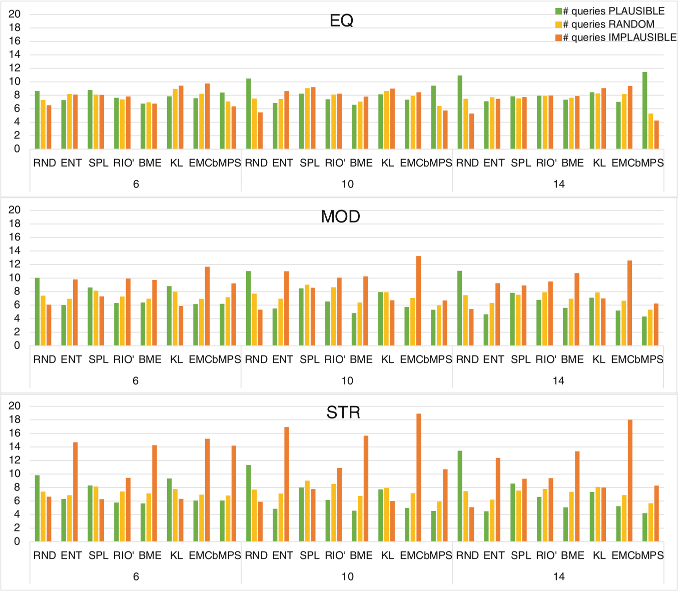

Fig. 1. Results for the University (U) case. (Color figure online)

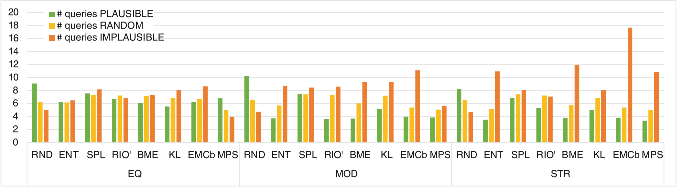

Fig. 2. Results for the MiniTambis (M) case with $ld = 10$ (F4). (Color figure online)

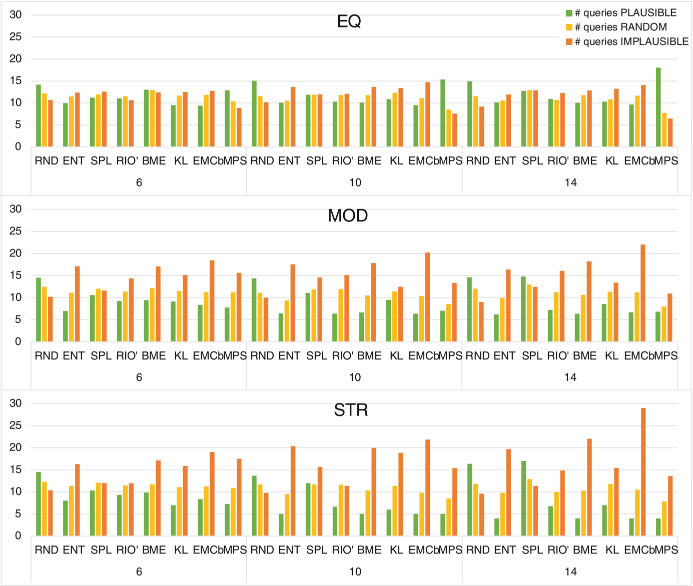

Fig. 3. Results for the Transportation (T) case. (Color figure online)

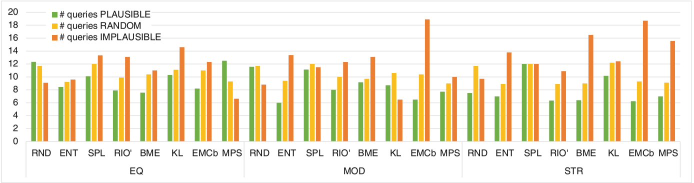

Fig. 4. Results for the Economy (E) case with $ld = 10$ (F4). (Color figure online)

The Plausible Cases: Throughout these cases we have the highest variation concerning the optimal QSM. Actually, all QSMs except for RND and SPL do appear as winners in certain cases. The distribution of the number of appearances as (or among) the best QSM(s) over all QSMs is displayed by Table 6 (rows 3 + 4). That, e.g., ENT is rather good in these cases and RND is no good choice (see also Figs. 1, 2, 3 and 4) is in agreement with the findings of [27]. However, we realize that BME is (among) the best QSMs more often than ENT. Comparing only these two, we find that BME outdoes ENT 7 times, ENT wins against BME 4 times, and they are equally good once. A reason for the strength of BME could be the fact that it will in most cases achieve only a minor bias towards one query outcome, as the maximization of the diagnoses elimination rate requires an as small as possible number of diagnoses with a probability sum >0.5. Thus, there is on the one hand a bias increasing the expected diagnoses invalidation rate, and on the other hand a near 50-50 outcome distribution implying a good entropy value. Unsurprisingly, if we sort the QSMs from most to least times being (among) the best based on Table 6 (rows 3 + 4), the resulting order coincides quite well with Eq. (1). In other words, in the plausible scenarios, probability-sensitive heuristics perform best.

Towards New QSMs/Meta-Heuristics. Exploiting the discussed results, one could endeavor to devise new QSMs that are superior to the investigated ones. For instance, in the implausible cases, only RND, MPS and KL occur as best QSMs. Thus, an optimal heuristic for these cases should likely adopt or unify selection principles of these three QSMs. One idea could be, e.g., to sample a few queries using RND and then choose the best one among them using (a weighted combination of) MPS and/or KL. Generally, one could use a meta heuristic that resorts to an appropriately (possibly dynamically re-)weighted sum of the QSM-functions (Table 1, 2nd column). Also, a QSM selecting queries based on a majority voting of multiple heuristics is thinkable, e.g., in Example 6 the query selected by such a QSM would be q_7 (cf. Table 3).

6 Conclusions and Future Work

Results of extensive evaluations on both classical and recently suggested query selection measures (QSMs) for test-driven ontology debugging (TOD) are presented. Main findings are: Using an appropriate QSM is essential, as otherwise TOD cost overheads of over 250% are possible. The one and only best QSM does not exist (or has not yet been found). Besides the size of the solution space of diagnoses, main factors influencing TOD cost are the bias in and the quality of the fault probability distribution, but not the ontology (debugging problem) as such or the size of the diagnoses sample used for query selection. Different QSMs prevail in the various probability distribution scenarios. Interestingly, the quite popular and frequently adopted entropy measure only manifested good (albeit not best) behavior in a single set of scenarios.

Future work topics include in-depth analyses of the (full) results and the design of new QSMs, e.g. meta-heuristics, based on the lessons learned. Moreover, machine learning techniques could be adopted to recommend optimal QSMs based on a classification of a debugging scenario wrt. the QSM-relevant factors we found. And, we plan to integrate the investigated QSMs into our Protégé ontology debugging plug-in [24].[5] From the application point of view – since the discussed techniques are not only applicable to ontologies, but to any monotonic knowledge representation formalism [16] – we intend to explore other use cases of the method. One example could be the adoption in the context of knowledge-based recommender systems [4] where model-based diagnosis is applied for relaxing user selection filters for avoiding empty result sets. Our gained insights could be profitably exploitable for guiding the relaxation process.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook. Cambridge University Press, Cambridge (2007)
2. De Kleer, J., Raiman, O.: Trading off the costs of inference vs. probing in diagnosis. In: IJCAI 1995, pp. 1736–1741 (1995)
3. Felfernig, A., Friedrich, G., Jannach, D., Stumptner, M.: Consistency-based diagnosis of configuration knowledge bases. Artif. Intell. **152**(2), 213–234 (2004)
4. Felfernig, A., Mairitsch, M., Mandl, M., Schubert, M., Teppan, E.: Utility-based repair of inconsistent requirements. In: Chien, B.-C., Hong, T.-P., Chen, S.-M., Ali, M. (eds.) IEA/AIE 2009. LNCS (LNAI), vol. 5579, pp. 162–171. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02568-6_17
5. Ferré, S., Rudolph, S.: Advocatus Diaboli – Exploratory Enrichment of Ontologies with Negative Constraints. In: ten Teije, A., et al. (eds.) EKAW 2012. LNCS (LNAI), vol. 7603, pp. 42–56. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33876-2_7
6. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P.F., Sattler, U.: OWL 2: The next step for OWL. JWS **6**(4), 309–322 (2008)
7. Hyafil, L., Rivest, R.L.: Constructing optimal binary decision trees is NP-complete. Inf. Process. Lett. **5**(1), 15–17 (1976)
8. Kalyanpur, A.: Debugging and repair of OWL ontologies. Ph.D. thesis, University of Maryland, College Park (2006)
9. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca-Grau, B.: Repairing unsatisfiable concepts in OWL ontologies. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 170–184. Springer, Heidelberg (2006). https://doi.org/10.1007/11762256_15
10. de Kleer, J., Raiman, O., Shirley, M.: One step lookahead is pretty good. In: Readings in Model-Based Diagnosis, pp. 138–142 (1992)

[5] http://isbi.aau.at/ontodebug.

11. de Kleer, J., Williams, B.C.: Diagnosing multiple faults. Artif. Intell. **32**(1), 97–130 (1987)
12. Moret, B.M.: Decision trees and diagrams. ACM Comput. Surv. (CSUR) **14**(4), 593–623 (1982)
13. Nikitina, N., Rudolph, S., Glimm, B.: Reasoning-supported interactive revision of knowledge bases. In: IJCAI 2011, pp. 1027–1032 (2011)
14. Rector, A., et al.: OWL pizzas: practical experience of teaching OWL-DL: common errors & common patterns. In: Motta, E., Shadbolt, N.R., Stutt, A., Gibbins, N. (eds.) Engineering Knowledge in the Age of the Semantic Web, EKAW 2004. LNCS, vol. 3257, pp. 63–81. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30202-5_5
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. Ph.D. thesis, University of Klagenfurt (2015). http://arxiv.org/pdf/1605.05950v1.pdf
17. Rodler, P.: Towards better response times and higher-quality queries in interactive knowledge base debugging. Technical report, University of Klagenfurt (2016). http://arxiv.org/pdf/1609.02584v2.pdf
18. Rodler, P.: On active learning strategies for sequential diagnosis. In: International Workshop on Principles of Diagnosis (DX 2017), pp. 264–283 (2018)
19. Rodler, P., Schekotihin, K.: Reducing model-based diagnosis to knowledge base debugging. In: International Workshop on Principles of Diagnosis (DX 2017), pp. 284–296 (2018)
20. Rodler, P., Schmid, W., Schekotihin, K.: A generally applicable, highly scalable measurement computation and optimization approach to sequential model-based diagnosis (2017). http://arxiv.org/abs/1711.05508
21. Rodler, P., Schmid, W., Schekotihin, K.: Inexpensive cost-optimized measurement proposal for sequential model-based diagnosis. In: International Workshop on Principles of Diagnosis (DX 2017), pp. 200–218 (2018)
22. Rodler, P., Shchekotykhin, K., Fleiss, P., Friedrich, G.: RIO: minimizing user interaction in ontology debugging. In: Faber, W., Lembo, D. (eds.) RR 2013. LNCS, vol. 7994, pp. 153–167. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39666-3_12
23. Roussey, C., Corcho, O., Vilches-Blázquez, L.M.: A catalogue of OWL ontology antipatterns. In: K-CAP 2009, pp. 205–206 (2009)
24. Schekotihin, K., Rodler, P., Schmid, W.: OntoDebug: interactive ontology debugging plug-in for Protégé. In: Ferrarotti, F., Woltran, S. (eds.) FoIKS 2018. LNCS, vol. 10833, pp. 340–359. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-90050-6_19
25. Schlobach, S., Huang, Z., Cornet, R., Harmelen, F.: Debugging incoherent terminologies. J. Autom. Reason. **39**(3), 317–349 (2007)
26. Settles, B.: Active Learning. Morgan and Claypool Publishers (2012)
27. Shchekotykhin, K., Friedrich, G., Fleiss, P., Rodler, P.: Interactive ontology debugging: two query strategies for efficient fault localization. JWS **12–13**, 88–103 (2012)

28. Shchekotykhin, K., Friedrich, G., Rodler, P., Fleiss, P.: Sequential diagnosis of high cardinality faults in knowledge-bases by direct diagnosis generation. In: ECAI 2014, pp. 813–818 (2014)
29. Stuckenschmidt, H.: Debugging OWL ontologies - a reality check. In: EON 2008, pp. 1–12 (2008)
30. Troquard, N., Confalonieri, R., Galliani, P., Penaloza, R., Porello, D., Kutz, O.: Repairing ontologies via axiom weakening (2017). http://arxiv.org/abs/1711.03430