The Empirical Robustness of Description Logic Classification

Rafael S. Gonçalves, Nicolas Matentzoglu, Bijan Parsia, and Uli Sattler

School of Computer Science, University of Manchester, Manchester, United Kingdom

Abstract. In spite of the recent renaissance in lightweight description logics (DLs), many prominent DLs, such as that underlying the Web Ontology Language (OWL), have high worst case complexity for their key inference services. Modern reasoners have a large array of optimization, tuned calculi, and implementation tricks that allow them to perform very well in a variety of application scenarios even though the complexity results ensure that they will perform poorly for some inputs. For users, the key question is how often they will encounter those pathological inputs in practice, that is, how robust are reasoners. We attempt to determine this question for classification of existing ontologies as they are found on the Web. It is a fairly common user task to examine ontologies published on the Web as part of their development process. Thus, the robustness of reasoners in this scenario is both directly interesting and provides some hints toward answering the broader question. From our experiments, we show that the current crop of OWL reasoners, in collaboration, is very robust against the Web.

1 Motivation

A serious concern about both versions 1 [12] and 2 [5] of the Web Ontology Language (OWL) is that the underlying description logics ($\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$) exhibit extremely bad worst case complexity (NEXPTIME and 2NEXPTIME) for their key inference services. While since the mid-1990s, highly optimized description logic reasoners have been exhibiting rather good performance in real cases, even in those more constrained cases there are ontologies (such as Galen) which have proved impossible to process for over a decade. Indeed, concern with such pathology stimulated a renaissance of research into tractable description logics with the $\mathcal{EL}$ family [1] and the DL Lite [4] family being incorporated as special "profiles" of OWL 2. However, even though the number of ontologies available on the Web has grown enormously since the standardization of OWL, it is still unclear how robust modern, highly optimized reasoners are to such input. Anecdotal evidence suggests that pathological cases are common enough to cause problems, however, systematic evidence has been scarce.

In this paper we investigate the question of whether modern, highly-optimized description logic reasoners are *robust* over Web input. The general intuition of a robust system is that it is *resistant to failure in the face of a range of input*. For any particular robustness determination, one must decide: *1)* the range of input, *2)* the functional or non-functional properties of interest, and *3)* what counts as failure. The instantiation of these parameters strongly influences robustness judgements, with the very same reasoner being highly robust under one scenario and very non-robust under another. For our current purposes, the key scenario is that an ontology engineer, using a tool like Protégé

[14], is inspecting ontologies published on the Web with an eye to possible reuse, and, as is common, they wish to classify the ontology using a standard OWL 2 DL reasoner as part of their evaluation. This scenario yields the following constraints: *1)* for input, we examine Web-based corpora, *2)* functional: acceptance (will the reasoner load and process the ontology); non-functional: performance (i.e., will the reasoner complete classification before the ontology engineer gives up), *3)* w.r.t. acceptance, failure means either rejecting the input or crashing while processing, and we might reasonably expect an engineer to wait up to 2 hours if the ontology seems "worth it". If a reasoner (or a set of reasoners) is successful for 90% of a corpus, we count that reasoner as robust over that corpus, with 95% and 99% indicating "strong" and "extreme" robustness. While these levels are clearly arbitrary (as is the timeout), they provide a framework to set expectations. Robustness under these assumptions does not ensure robustness under other assumptions (e.g., over subsets of these ontologies as experienced during development or over a more stringent time constraint), yet they are challenging enough that it was unclear to us *ex ante* whether any reasoners would be robust for any corpus. In fact, we find that the reasoners are robust or near robust for most of the cases we examine including for lower timeouts. More significantly, if we take the best result for each ontology (which represents a kind of "meta-reasoner", where our test reasoners are run in parallel), then the *set* of reasoners is extremely robust over all corpora. Thus, in a fairly precise, if limited, sense, we demonstrate that $\mathcal{SHOIQ}$ and $\mathcal{SROIQ}$ are practical description logics.

2 Materials & Methods

For our input data, we gathered three sets of ontologies from the Web — all versions of the NCI Thesaurus (NCIt), ontologies in the NCBO Bioportal repository, and the results of a Web crawl, each with fundamentally different characteristics.

The NCIt has been continuously developed and published in monthly versions since 2003. The NCIt archive[1] contains 106 versions parseable by the OWL API [10],[2] from release 02.00 (October 2003) through to release 12.11d (November 2012) ranging in size from 49,475 to 133,900 logical axioms and in expressivity from $\mathcal{ALE}$ to $\mathcal{SH}(D)$. The NCIt team is a fairly stable, closed team of about 20 ontology developers who use a highly regimented process and, at least since 2006, have incorporated OWL reasoners in their tools chain (namely FaCT++ and Pellet). The NCIt is large and easily accessible, thus has been an informal benchmark for reasoner developers. Additionally, the NCI has funded various infrastructure projects, including improvements to reasoners. Thus, we might reasonably expect that reasoners are robust w.r.t. this corpus, both because the NCI team may be tuning their ontology to the available reasoners (though the fact that they fund improvements suggests not), and because reasoner developers are tuning for NCIt.

The NCBO Bioportal is a Web based repository for health care and life science ontologies. We use a snapshot of (publicly downloadable ontologies from) the BioPortal

[1] http://evs.nci.nih.gov/ftp1/NCI_Thesaurus

[2] http://owlapi.sourceforge.net

repository from November 2012, consisting of 292 OWL and OBO parseable ontologies. The average number of logical axioms in the corpus is 28,439 (total: 8,190,504 and median: 979 axioms), and 89 of these ontologies contain named individuals. 4 ontologies contained no logical axioms at all and thus were discarded. In expressivity, the ontologies range from the inexpressive $\mathcal{AL}$ DL to the very expressive $\mathcal{SROIQ}$. The ontologies are developed and used in a wide range of largely unrelated projects for a variety of purposes using a variety of tools. While Bioportal has received some attention from the research community, it is not yet a standard target for reasoner developers.

The third corpus, obtained by a short Web crawl and fuelled by a high number of seeds from Swoogle, Google and ontology repositories on the Web, was collected in November 2012. We picked a random sample of 822 ontologies, out of which 145 contained no logical axioms at all and thus were discarded, leaving 677 ontologies for our experiment. The average number of logical axioms is 2,405 (total: 1,628,207 and median: 57), and the expressivity ranges from $\mathcal{AL}$ to $\mathcal{SRIQ}$. These ontologies span a wide range of subjects and are completely uncontrolled with respect to their origin. Perhaps not surprisingly, there are fewer axioms overall and on average, with half of the ontologies containing under 60 axioms. This may reflect less commitment to the ontologies than we see in the more curated set. However, there is no reason to think that the reasoners have been specially tuned to these ontologies and, given the worst case complexity of the logics, even small ontologies are a potential pathological case. Thus, it is unclear what the rational robustness expectation is for this set.

We selected four reasoners for testing based on the following criteria: *a)* coverage of all of OWL 2, *b)* freely available for download, *c)* native support for the OWL API, and finally *d)* based on sound, complete and terminating algorithms. As such, the chosen reasoners are Pellet [19], HermiT [18], FaCT++ [20], and JFact. We excluded, e.g., the RacerPro [9], CB [13], and KAON2[3] reasoners due to their lack of coverage for all OWL 2 features, and no native support for the OWL API. Finally, we did not consider approximate (either unsound or incomplete) reasoners, such as TrOWL [16], so that we can compare classification results between reasoners, and because we feel that approximation is generally only considered in cases where sound and complete reasoners fail.

For all our experiments we use the current 2013 reasoner versions, namely, Pellet v2.3.0, HermiT v1.3.6, FaCT++ v1.6.1 and JFact v1.0. However, since NCIt performance has been studied previously [7], we decided to compare the current reasoner versions with the versions used in the 2011 study, namely Pellet v2.2.2, HermiT v1.3.3, FaCT++ v1.5.3 and JFact v0.2, in order to test how much tuning to NCIt occurs.

As mentioned earlier, we set the classification timeout tp 2 hours per ontology-reasoner pair. From a scenario perspective, 2 hours is rather generous – many ontologists will give up much sooner than that. However, 2 hours gives us an idea of which "hard" ontologies are clearly in striking distance, without making completing the experiments infeasible. In the presentation below, we examine a tighter timeout (and thus harder robustness criterion) of about 100 seconds. The main experiment machine has an Intel Quad-Core Xeon 3.2GHz processor with 32GB DDR3 RAM. A second experiment involving solely the NCIt (and both reasoner sets) was performed on a machine

[3] `http://kaon2.semanticweb.org`

with an Intel Dual-Core i7 2.7GHz processor, with 16GB DDR3 RAM. All tests were run on Mac OS X 10.7.5, using Java v1.7 and the OWL API v3.4.1.

The test corpora, experiment results, and reasoners used are available from `http://sites.google.com/site/reasonerbenchmark`.[4]

3 Results

In all experiments we categorise ontology classification times into the following bins: Very Easy ($\leq$ 1 second), Easy (1-10 seconds), Medium (10-100 seconds), Hard (100-1000 seconds), and Very Hard ($>$1000 seconds). We denote "Impatient Robustness" as a measure of how many ontologies terminate in an acceptable time for most users, i.e., ontologies in the Medium bin or below. Throughout this section we use "Best Combo" as the best of all of 4 reasoners' results (i.e., fastest time), and, analogously, "Worst Combo" as the worst.

3.1 NCI Thesaurus

In this experiment we test both 2011 and 2013 reasoner versions sets, and compare the performance behaviour of each reasoner. The classification times for both reasoner sets are shown in Figure 1. Using the reasoner versions from 2011, and taking into account those ontologies that *all* reasoners managed to process and classify, FaCT++ is on average the fastest of all 4 reasoners, taking 14.7 seconds per version. JFact comes second, with an average of 22.9 seconds per version, while Pellet is the third fastest, taking on average 36.5 seconds per version, and finally HermiT is the slowest, with 150 seconds per version (see Table 1).

When switching to the 2013 reasoner sets the performance winner remains FaCT++, with an average of 19.2 seconds per ontology. However in second place now comes Pellet, taking 61.6 seconds on average, in third HermiT with an average of 174 seconds, and finally JFact taking 180 seconds on average per ontology. Notice that from 2011 to 2013 there was a significant improvement in JFact's robustness, with far fewer errors. Similarly Pellet has less errors in its most recent version, and the performance is superior to the 2011 version. FaCT++ and HermiT's performance slightly decreased from 2011 to 2013 on this corpus, though not nearly as much as JFact, possibly because in its most recent version JFact is able to process the more recent versions of the NCIt.

Overall, FaCT++ is the most robust reasoner for the NCIt corpus, having no errors in either 2011 or 2013 versions (see Tables 1 and 2). Furthermore, it is the fastest performing reasoner across nearly all versions. The least robust is, interestingly, FaCT++'s port to the Java language: JFact, due to the high number of errors reported. Though there is improvement from 2011 to 2013, this "young" reasoner is still not as fast as FaCT++. The reasoner errors encountered throughout the NCIt were, by Pellet: "OutOfMemory" errors, by HermiT: "StackOverflow", and finally by JFact: "IllegalArgument".

[4] Full set of crawled ontologies is available upon request.

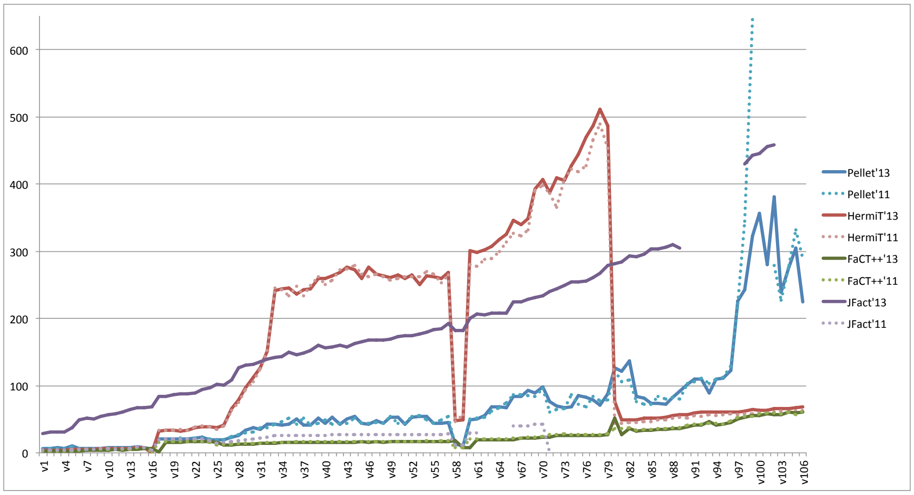


Fig. 1. Comparison of classification times between the 2011 reasoner version set (suffixed '11) and the 2013 set (suffixed '13) over the NCIt corpus (*y*-axis: time in seconds, *x*-axis: version number).

	Pellet	HermiT	JFact	FaCT++	Best Combo	Worst Combo
Very Easy	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Easy	16 (15%)	15 (14%)	16 (15%)	18 (17%)	18 (17%)	15 (14%)
Medium	70 (66%)	42 (40%)	52 (49%)	88 (83%)	88 (83%)	15 (14%)
Hard	18 (17%)	48 (45%)	0 (0%)	0 (0%)	0 (0%)	37 (35%)
Very Hard	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Timeout	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Errors	2 (2%)	1 (1%)	38 (36%)	0 (0%)	0 (0%)	39 (37%)
Impatient Robustness	81%	54%	64%	100%	100%	28%
Overall Robustness	98%	99%	64%	100%	100%	63%

Table 1. Binning of the NCIt corpus according to performance (2011 reasoner versions sets).

	Pellet	HermiT	JFact	FaCT++	Best Combo	Worst Combo
Very Easy	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Easy	16 (15%)	15 (14%)	0 (0%)	19 (18%)	19 (18%)	0 (0%)
Medium	71 (67%)	42 (40%)	24 (23%)	87 (82%)	87 (82%)	23 (22%)
Hard	19 (18%)	48 (45%)	70 (66%)	0 (0%)	0 (0%)	70 (66%)
Very Hard	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Timeout	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Errors	0 (0%)	1 (1%)	12 (11%)	0 (0%)	0 (0%)	13 (12%)
Impatient Robustness	82%	54%	23%	100%	100%	22%
Overall Robustness	100%	99%	89%	100%	100%	88%

Table 2. Binning of the NCIt corpus according to performance (2013 reasoner versions sets).

3.2 NCBO BioPortal

In the snapshot of BioPortal, out of all 288 non-empty ontologies, 9 ontologies are inconsistent and there are 234 that *all* reasoners manage to classify within the timeout (see Table 3). Out of those ontologies where all reasoners completed classification, FaCT++ was on average the fastest (2.9 seconds per ontology), followed by JFact (5.9 seconds), HermiT (9.8 seconds), and finally Pellet (16.7 seconds). However, in terms of robustness, our results show that Pellet is the most robust of all reasoners, only failing to handle 9 ontologies, while HermiT, the least robust, fails to classify 27 ontologies. FaCT++ and JFact fail to handle 24 and 25 ontologies respectively (see Table 4 for more details regarding errors).

Generally, Pellet is not only the most robust reasoner for BioPortal, with fewer errors, but also exhibits fast performance on a high number of ontologies. However, it does have the most timeouts; but note that some of these were on ontologies that other reasoners threw an error on. The remaining 3 reasoners are very close to each other performance and robustness-wise, HermiT with less timeouts but more errors than JFact and FaCT++, and slower performance. Thus HermiT is the least robust reasoner for BioPortal.

	Pellet	HermiT	JFact	FaCT++	Best Combo	Worst Combo
Very Easy	190 (66%)	170 (59%)	184 (64%)	218 (76%)	236 (82%)	152 (53%)
Easy	56 (19%)	61 (21%)	58 (20%)	24 (8%)	28 (10%)	58 (20%)
Medium	10 (3%)	15 (5%)	8 (3%)	7 (2%)	11 (4%)	10 (3%)
Hard	4 (1%)	4 (1%)	2 (1%)	2 (1%)	4 (1%)	2 (1%)
Very Hard	6 (2%)	3 (1%)	0 (0%)	3 (1%)	4 (1%)	2 (1%)
Timeout	13 (5%)	8 (3%)	11 (4%)	10 (3%)	5 (2%)	15 (5%)
Errors	9 (3%)	27 (9%)	25 (9%)	24 (8%)	0 (0%)	49 (17%)
Impatient Robustness	89%	85%	87%	86%	95%	76%
Overall Robustness	92%	88%	88%	88%	98%	78%

Table 3. Binning of the BioPortal corpus according to performance.

Error	Pellet	HermiT	JFact	FaCT++
StackOverflow	2	0	1	0
OutOfMemory	1	1	2	0
UnsupportedDatatype	0	13	4	14
InternalReasoner	2	0	1	0
IllegalArgument	0	12	16	6
MalformedLiteral	0	1	0	0
ConcurrentModification	3	0	0	0
Reasoner crashed	0	0	0	4
IndexOutOfBounds	1	0	1	0
Total Errors	9	27	25	24

Table 4. Errors and exceptions that occurred during classification of BioPortal ontologies.

3.3 Web Crawl Corpus

Out of the 677 non-empty ontologies from the Web crawl corpus, *all* reasoners completed classification of 560 of them. In these 560, Pellet was the fastest reasoner on average (0.5 seconds per ontology), followed by FaCT++ (1.5 seconds), HermiT (3.1 seconds), and finally JFact (6.2 seconds). In terms of robustness, Pellet is, again, the most robust, having only thrown errors on 17 ontologies (see Table 5). It is also the reasoner with most timeouts, but again, several times where other reasoners threw errors. FaCT++ and HermiT both have a high number of errors, while, curiously, JFact did much better on that front in this corpus. In Table 6 the errors found across the corpus are broken down.

	Pellet	HermiT	JFact	FaCT++	Best Combo	Worst Combo
Very Easy	597 (88%)	536 (79%)	557 (82%)	566 (84%)	642 (95%)	493 (73%)
Easy	44 (6%)	36 (5%)	45 (7%)	12 (2%)	26 (4%)	44 (6%)
Medium	2 (0%)	8 (1%)	11 (2%)	0 (0%)	3 (0%)	12 (2%)
Hard	1 (0%)	1 (0%)	4 (1%)	5 (1%)	2 (0%)	3 (0%)
Very Hard	0 (0%)	1 (0%)	1 (0%)	1 (0%)	0 (0%)	1 (0%)
Timeout	16 (2%)	6 (1%)	5 (1%)	5 (1%)	4 (1%)	10 (1%)
Reasoner Errors	17 (3%)	89 (13%)	54 (8%)	88 (13%)	0 (0%)	114 (17%)
Impatient Robustness	95%	86%	91%	85%	99%	81%
Overall Robustness	95%	86%	91%	86%	99%	82%

Table 5. Binning of the Web crawl corpus according to performance.

Error	Pellet	Hermit	JFact	FaCT++
StackOverflow	13	0	0	0
OutOfMemory	2	0	2	0
NullPointer	0	0	36	0
UnloadableImport	0	1	1	1
ClassCast	0	0	1	0
UnsupportedDatatype	0	81	1	86
Datatype constraint	2	0	0	0
IllegalArgument	0	3	5	0
MalformedLiteral	0	2	0	0
ReasonerInternal	0	0	8	1
UnsupportedFacet	0	2	0	0
Total	17	89	54	88

Table 6. Errors and exceptions that occurred during classification of the Web crawl ontologies.

Overall Pellet is the most robust and fastest (among ontologies that could be classified by all reasoners) reasoner for this corpus, followed closely by JFact, both in terms of robustness and performance. The least robust reasoners for the Web crawl corpus are FaCT++ and HermiT, with 88 and 89 errors, respectively. However, HermiT performed slightly better on the lower bins, while FaCT++ was clearly the slowest in this corpus.

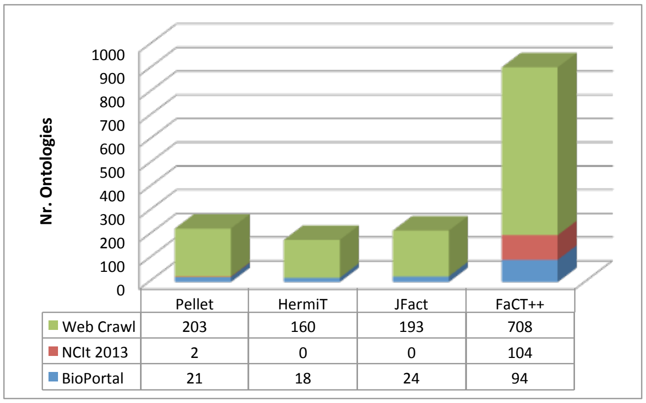
4 Discussion

Overall we have processed a total of 1,071 ontologies, the largest such reasoner benchmark, having found that amongst the 4 tested reasoners Pellet is the most robust of all (see Table 7). Surprisingly, Pellet is followed by JFact on our robustness test, due to having far less errors than FaCT++. HermiT and FaCT++ have the same overall robustness, but FaCT++ has less errors and higher impatient robustness.

	Pellet	HermiT	JFact	FaCT++	Best Combo	Worst Combo
Very Easy	787 (73%)	706 (66%)	741 (69%)	784 (73%)	878 (82%)	645 (60.2%)
Easy	116 (11%)	112 (10%)	103 (10%)	55 (5%)	73 (7%)	102 (9.5%)
Medium	83 (8%)	65 (6%)	43 (4%)	94 (9%)	101 (9%)	45 (4.2%)
Hard	24 (2%)	53 (5%)	76 (7%)	7 (1%)	6 (1%)	75 (7.0%)
Very Hard	6 (1%)	4 (0%)	1 (0%)	4 (0%)	4 (0%)	3 (0.3%)
Timeout	29 (3%)	14 (1%)	16 (1%)	15 (1%)	9 (1%)	25 (2.3%)
Errors	26 (2%)	117 (11%)	91 (8%)	112 (10%)	0 (0%)	176 (16.4%)
Total (excl. Errors)	1016	940	964	944	1062	870
Total (incl. Errors)	1071	1071	1071	1071	1071	1071
Impatient Robustness	92%	82% [90%]	83%	87% [96%]	98%	74% [87%]
Overall Robustness	95%	88% [96%]	90%	88% [97%]	99%	81% [96%]

Table 7. Binning of all three corpora: BioPortal, NCIt (2013), and Web crawl. Under robustness rows, values in square brackets indicate robustness w.r.t. OWL 2 alone.

While Pellet is the most robust reasoner, we urge some caution in that reading. In particular, this does not mean that Pellet will always do best or even perform reasonably. In fact, it may timeout where other reasoners finish reasonably fast. The set of reasoners (taken together and considering the best results) is extremely robust across the board (for each reasoner's contribution to the best case reasoner, see Figure 2). Thus, we have strong empirical evidence that the *ontologies* on the Web do not supply many *in principle* intractable cases, but only cases which are difficult for particular reasoners.



	Pellet	HermiT	JFact	FaCT++
🟩 Web Crawl	203	160	193	708
🟥 NCIt 2013	2	0	0	104
🟦 BioPortal	21	18	24	94

Fig. 2. Number of times that each reasoner equals the best case, for each corpus.

Note that FaCT++ and JFact fail to process several ontologies due to poor support for OWL datatypes, particularly datatypes not specified in the OWL 2 datatype map; both of these reasoners, as well as HermiT, have little support for OWL 1 datatypes. By removing the non OWL 2 datatype errors, we would end up with FaCT++ being the most robust w.r.t. OWL 2, followed by HermiT and Pellet. From Figure 2 we see that FaCT++ outperforms other reasoners on many occasions, but, due to the high number of errors thrown, its robustness w.r.t. our input data is not nearly at the same level as its performance.

The 9 ontologies which no reasoner classified within the timeout range in expressivity between $\mathcal{ALEHIF}+$ and $\mathcal{SRIQ}$. Their average number of logical axioms is 56,179; the minimum is 341 axioms - $\mathcal{SRIQ}$ ontology, maximum 379,734 axioms - $\mathcal{SR}$ ontology, and median 17,385 axioms - $\mathcal{SHIF}$ ontology.

It is clear that deriving a sensible ranking even simply using average or total time is not straightforward. Our results have rather strong implications for reasoner experiments, especially those purporting to show the advantages of an optimisation or a technique or an implementation: The space is very complex and it is very easy to simultaneously generate a biased sample for one system and against another. Even simple, seemingly innocuous things like timeouts and classification failures require tremendous care in handling. If results are going to be meaningful across papers we need to converge on experimental inputs, methods, and reporting forms.

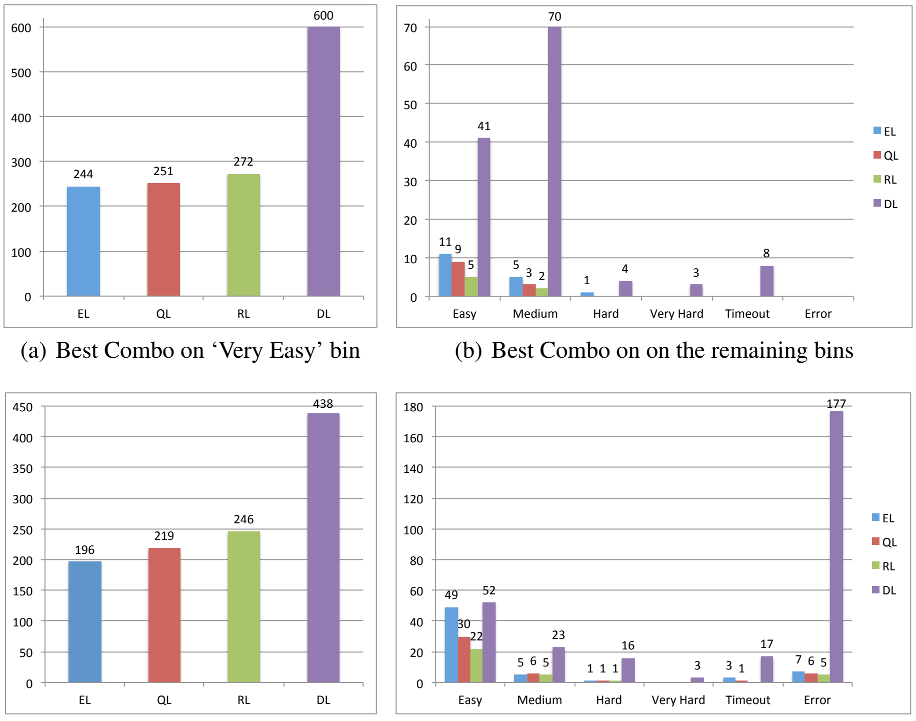
Finally, in order to get an overall picture of how these robustness measurements relate to the OWL profile in which ontologies fit into, we: divide our ontologies into their corresponding OWL profile, and match them with the observed performance bin of the Best and Worst Combo reasoners. This is displayed in Figure 3.

Since there is an overlap between the EL, RL and QL profiles of OWL 2, some ontologies are counted in more than one such bin, meaning that the total number of ontologies in Figure 3 does not add up to the number of ontologies in our corpus. However, the ontologies contained in the DL profile bin are exclusive, i.e., an ontology in the EL profile is not counted again within the DL profile. Note that, even though ontologies in the EL, RL and QL profiles of OWL are typically in the easier bins, there are some which are deemed hard, time out, or even result in error.

5 Related Work

There is extensive work in benchmarking reasoners, some of which focuses purely on either classification or (conjunctive) query answering (e.g., [8,17]). Generally, previous reasoner benchmarks used much smaller and rather *ad hoc* data sets, in some cases using artificial data. For the purposes of this paper, we focus solely on work involving the classification task, particularly using realistic rather than artificially-generated test data.

The Pellet reasoner was evaluated, in [19], with a corpus of 9 ontologies, presenting the average of 10 independent runs of a reasoning task - the tasks under test being consistency checking, classification and realization. Additionally, the authors compare Pellet against FaCT++ and RacerPro in terms of classification time only, using the DL benchmark test suite described in [11]. The experiment showed that Pellet was not as efficient as FaCT++ or RacerPro in many, but not all, cases.



(a) Best Combo on ‘Very Easy’ bin

(b) Best Combo on on the remaining bins

(c) Worst Combo on ‘Very Easy’ bin

(d) Worst Combo on the remaining bins

Fig. 3. Number of ontologies in each OWL 2 profile displayed according to the performance profile of the Best and Worst Combo reasoners. On the left-hand side (Figures 3(a) and 3(c)) we show the OWL profile distribution of the ontologies in the ‘Very Easy’ performance bin, as it is the most densely populated bin. While on the right-hand side (Figures 3(b) and 3(d)) the remaining performance bins.

In [6] the authors present a system for comparing reasoners both in terms of performance and correctness of classification results. Four reasoners are put to test: FaCT++, Pellet, KAON2 and RacerPro, over a corpus of 172 naturally occurring ontologies, out of which only 31 were either in or more expressive than $\mathcal{ALC}$. The benchmark results show that Pellet was the most robust reasoner, with FaCT++ a close second, being able to process, respectively, 143 and 137 ontologies. In terms of classification time, the authors state that “there is no clear winner”, due to considerable fluctuation of reasoner performance across ontologies.

The evaluation of the HermiT reasoner [18] was carried out against the FaCT++ and Pellet reasoners, using a corpus of ontologies derived from the Gardiner data set [6], the Open Biological Ontologies (OBO) Foundry,[5] and finally, several versions of the GALEN ontology. The result was that HermiT outperforms the other reasoners in the majority of tested ontologies.

[5] `http://obofoundry.org`

In [3] the authors carry out a benchmark of ontologies derived from the Watson repository.[6] Out of the 6,224 ontologies in Watson, only 3,303 were parseable by both Swoop and the KAON2 tools. These were then classified into 4 bins according to their expressivity; RDFS(DL), OWL DLP, OWL Lite, and OWL DL. From these bins, the authors picked 1 representative per bin, according to its popularity in previous benchmarks. The test itself involved the reasoners HermiT, Pellet, RacerPro, KAON2, OWLIM and Sesame, where the classification performance results show that HermiT was fastest in 3 out 4 cases, OWLIM being the fastest in the RDFS(DL) representative.

The author of [15] performs a benchmark of the Pellet, FaCT++ and Racer reasoners, though using different interfaces (FaCT++ used DIG at the time) - thus the results are not directly comparable. This benchmark was carried out using a corpus of 135 OWL ontologies from Schemaweb.[7] The experiment showed that FaCT++ was the fastest (excluding timeouts) and the most robust, since it processed the most ontologies without timing-out or aborting (due to errors unrevealed by the author).

The benchmark carried out in [2] compared the KAON2, Pellet, Racer, HermiT and FaCT++ reasoners, against 50 naturally occurring ontologies. However, in the paper, the authors focus only on a few examples; Racer was fastest on the *Wine* ontology, HermiT on *DeepTree*, FaCT++ on the NCI Thesaurus, and HermiT on *GALEN*.

6 Future Work

In this paper, we did not have space to discuss whether there is a performance/size or performance/expressivity correlation. By and large, our analysis shows that there is a roughly linear correlation between performance and size, and no correlation with expressivity.

Due to the large size of the Web crawl corpus, we resorted to sampling in order to obtain results in time. Though we have tested large enough samples to attain statistical significance, we hope to complete processing all ontologies in said corpus in the near future. For the purposes of this paper we limited our attention to classification, but could easily extend our benchmarking to other inference problems, even to non-standard ones such as justification finding. We also intend to tackle the vast task of identifying promising correlations between features of ontologies and their reasoning difficulty.

To address the difficulties in stable, cross-experiment comparison and interpretation, we propose to establish a comprehensive benchmark which is updated yearly. To facilitate rapid experimentation, we will provide canonical stable random samples so that experimenters can provide a comparable baseline, even if for scientific reasons they must also investigate other inputs. We will also make our test framework and computing platform available, re-running all the experiments we can gather in the prior year to provide systematic review and replication of results.

[6] http://watson.kmi.open.ac.uk/WatsonWUI/

[7] http://schemaweb.info/

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the EL envelope. In: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05) (2005)
2. Babik, M., Hluchy, L.: A testing framework for OWL-DL reasoning. In: Proc. of the Int. Conf. on Semantics, Knowledge and Grids (SKG-08) (2008)
3. Bock, J., Haase, P., Ji, Q., Volz, R.: Benchmarking owl reasoners. In: Proc. of the Int. Workshop on Advancing Reasoning on the Web: Scalability and Commonsense (ARea-08) (2008)
4. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. J. of Automated Reasoning 39(3), 385–429 (2007)
5. Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P.F., Sattler, U.: OWL 2: The next step for OWL. J. of Web Semantics (2008)
6. Gardiner, T., Tsarkov, D., Horrocks, I.: Framework for an automated comparison of description logic reasoners. In: Proc. of the 5th Int. Semantic Web Conf. (ISWC-06) (2006)
7. Gonçalves, R.S., Parsia, B., Sattler, U.: Analysing the evolution of the NCI thesaurus. In: Proc. of the 24th IEEE Int. Symposium on Computer-Based Medical Systems (CBMS-11) (2011)
8. Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. J. of Web Semantics 3(2-3), 158–182 (2005)
9. Haarslev, V., Möller, R.: RACER system description. In: Proc. of the 1st Int. Joint Conf. on Automated Reasoning (IJCAR-01). Lecture Notes in Artificial Intelligence, vol. 2083. Springer-Verlag (2001)
10. Horridge, M., Bechhofer, S.: The OWL API: A Java API for working with OWL 2 ontologies. In: Proc. of the 6th Int. Workshop on OWL: Experiences and Directions (OWLED-09) (2009)
11. Horrocks, I., Patel-Schneider, P.F.: DL systems comparison. In: Proc. of the 11th Int. Workshop on Description Logics (DL-98) (1998)
12. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. J. of Web Semantics 1(1), 7–26 (2003)
13. Kazakov, Y.: Consequence-driven reasoning for Horn $\mathcal{SHIQ}$ ontologies. In: Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI-09) (2009)
14. Knublauch, H., Fergerson, R.W., Noy, N.F., Musen, M.A.: The Protégé OWL plugin: An open development environment for semantic web applications. In: Proc. of the 3rd Int. Semantic Web Conf. (ISWC-04) (2004)
15. Pan, Z.: Benchmarking DL reasoners using realistic ontologies. In: Proc. of the 1st Int. Workshop on OWL: Experiences and Directions (OWLED-05) (2005)
16. Ren, Y., Pan, J.Z., Zhao, Y.: Soundness preserving approximation for tbox reasoning. In: Proc. of the 24th AAAI Conf. on Artificial Intelligence (AAAI-10) (2010)
17. Sattler, U., Motik, B.: A comparison of reasoning techniques for querying large description logic aboxes. In: Proc. of the 13th Int. Conf. on Logic for Programming and Automated Reasoning (LPAR-06) (2006)
18. Shearer, R., Motik, B., Horrocks, I.: HermiT: A highly-efficient OWL reasoner. In: Proc. of the 5th Int. Workshop on OWL: Experiences and Directions (OWLED-08EU) (2008)
19. Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. of Web Semantics 5(2), 51–53 (2007)
20. Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. In: Proc. of the 3rd Int. Joint Conf. on Automated Reasoning (IJCAR-06) (2006)