A Rigorous Characterization of Classification Performance – A Tale of Four Reasoners

Yong-Bin Kang[†], Yuan-Fang Li[†], Shonali Krishnaswamy[†,§]

[†] Faculty of IT, Monash University, Australia
{yongbin.kang,yuanfang.li,shonali.krishnaswamy}@monash.edu
[§] Institute for Infocomm Research, A*STAR, Singapore

Abstract. A number of ontology reasoners have been developed for reasoning over highly expressive ontology languages such as OWL DL and OWL 2 DL. Such languages have, as a consequence of high expressivity, high worst-case complexity. Therefore, reasoning tasks such as classification sometimes take considerable time on large and complex ontologies. In this paper, we carry out a comprehensive comparative study to analyze classification performance of four widely-used reasoners, FaCT++, HermiT, Pellet and TrOWL, using a dataset of over 300 real-world ontologies. Our investigation on correlating reasoner performance with ontology metrics using machine learning techniques also provides additional insights into the hardness of individual ontologies.

1 Introduction

Ontology reasoning tasks such as classification and consistency checking are fundamental to semantics-enabled applications. Very expressive ontology languages that can model complex domain knowledge have been designed and are widely used in a number of domains. Such languages include OWL DL [14] and its successor, OWL 2 DL [11]. High expressivity, however, incurs high computational complexity. For the core reasoning tasks of classification and consistency checking, OWL DL is NExpTime-complete, while OWL 2 DL is 2NExpTime-complete. Hence, terminological reasoning over such languages is a challenging task, especially for very large ontologies.

Highly efficient TBox reasoning algorithms such as those based on tableaux [4, 15] and hypertableaux [19] have been proposed to tackle this formidable problem. Various optimization techniques such as absorption, backtracking and blocking have been developed [13] to reduce search space, therefore speeding up the processing and reducing memory footprint. Based on these algorithms, a number of efficient ontology reasoners such as FaCT++ [26], HermiT [19], Pellet [23] and TrOWL [25] have been implemented. These reasoners can handle some very large ontologies such as GALEN, Gene Ontology and NCI Thesaurus Ontology. However, it has also been pointed out that further studies are still needed for improving terminological reasoning [9].

In a lot of situations such as in the mobile context [24], it is very valuable to obtain a (rough) estimate of reasoning performance before reasoning is actually carried out. Although theoretical worst-case complexity has been established for these languages, such complexity is not necessarily a reliable indication of real-world, typical-case performance. Part of the reason is that different reasoners implement different algorithms and optimization techniques, hence they may have widely different performance for a same ontology. In other words, the hardness of reasoning on individual ontologies is a product of the intrinsic characteristics of the ontologies (i.e., metrics [28]) and that of the reasoner employed.

Therefore, we believe it is of both theoretical and practical importance to adequately measure, benchmark and characterize performance of different reasoners. Many existing works on (TBox) reasoner benchmarking [20, 5, 7, 9, 10] have used relatively small to medium-sized datasets, which do not provide sufficient grounds for rigorously analysis of performance characteristics. These works also only focused on comparing and benchmarking performance of different reasoners – they did not provide insights into such performance.

In this paper, we attempt to conduct a rigorous and comprehensive study that characterizes performance of the above four reasoners, for the task of ontology classification, on a set of over 300 ontologies of varying sizes and hardness. We also study the relationship of the hardness of individual ontologies and their intrinsic syntactic and structural metrics [28] by applying a machine learning approach. Our preliminary results are very encouraging, showing a high accuracy of correctly predicting (discretized) performance of all the four reasoners.

2 Background and Related Work

Tremendous progress has been made in recent years in designing and implementing highly optimised inference algorithms and reasoners. Tableau- and hypertableau-based algorithms [3, 6, 19] have dominated DL inference research and many reasoners are based on these algorithms, including FaCT++ [26], Pellet [22] and HermiT [21]. With the introduction of OWL 2 and its profiles, other approaches, including completion rule-based and consequence-based algorithms have been developed to tackle inference problems on less expressive DLs such as $\mathcal{EL}^{++}$ (the OWL 2 EL profile) and DL-Lite (the OWL 2 QL profile), for which polynomial-time algorithms exist for standard DL inference tasks such as subsumption checking [1, 8]. Reasoners including `CEL` [2], CB [16] and `Snorocket` [17] are based on this approach. TrOWL [25] is an inference infrastructure that takes a hybrid approach: it applies syntactic and semantic approximation to transform OWL 2 DL ontologies to less expressive profiles (QL and EL) for different reasoning tasks, and it uses a variety of underlying reasoners for different languages.

Quite a few works have been done on benchmarking ontology reasoners. Earlier works primarily focused on OWL 1 and DAML+OIL ontologies. Bock et al. [7] benchmarked the time performance of 5 reasoners, KAON2, OWLIM, Pellet, RacerPro and Sesame, over a dataset generated from four small ontologies by varying the number of ABox assertions. Two reasoning tasks were evaluated: classification and conjunctive query answering. Because of the size of the ontologies, the majority of reasoners achieve a subsecond response time for classification on the four ontologies. On the other hand, they exhibit a more varying behavior for conjunctive query answering. Pan [20] compared three reasoners, FaCT++, Pellet and RacerPro, on a dataset of 135 (OWL 1) ontologies for the task of classification, and commented on the relative strengths and weaknesses of the reasoners. These ontologies are relatively small too: with an average of 43.7 classes and 19.3 relations per ontology. Gardiner [10] et al. also compared four reasoners, FaCT++, KAON2, Pellet and RacerPro, on 172 (OWL 1) ontologies. Their experiments showed that different reasoners have different characteristics, but did not discuss these differences in detail.

A new benchmarking framework based on justifications has recently been proposed by Bail et al. [5]. Justifications are small minimal subsets of logical axioms and assertions sufficient for an entailment to hold. The authors argued that a justification-based, but not classification-based, benchmarking approach provides better fault isolation capabilities and is useful in reasoner development.

More recently, Dentler et al. [9] conducted a comprehensive comparative study of three dimensions of eight reasoners, CB, CEL, FaCT++, HermiT, Pellet, RacerPro, Snorocket and TrOWL, that support the OWL 2 EL profile. A number of TBox reasoning tasks are performed on three large OWL 2 EL ontologies (Gene Ontology, NCI Thesaurus and SNOMED CT) and it was observed that the reasoners exhibit a significant difference in performance, and that further research is required to better understand this phenomenon.

In the SEALS project[1], the Storage and Reasoning Systems Evaluation Campaign 2010 aimed at evaluating DL-based reasoners. In the evaluation, the performance of three reasoners FaCT++, HermiT, and jcel were measured and compared in terms of a suite of standard inference services such as classification, class/ontology satisfiability, and logical entailment. This evaluation results in a framework revealing a good performance comparison of different reasoners. However, it does not seem to tackle the problem of performance prediction. Hence, our work presented here is complementary to the SEALS project.

3 Methodology

The principal aims of this paper are (1) to benchmark the performance of reasoning tasks of a number of reasoners over a large and diverse dataset, and (2) to experimentally determine whether a combination of ontology metrics can be leveraged to effectively predict the response time for specific reasoning tasks. Thus there are four dimensions which need to be considered:

Reasoning task - For our evaluation, we focus on *classification*. Classification is the process of making all class subsumption relations explicit in an ontology and it is one of fundamental TBox reasoning tasks. Another main reasoning task, consistency checking, is not chosen because of a pragmatic reason: that different reasoners perform consistency checking at different times. It is sometimes performed together with ontology loading in some reasoners, while some

[1] `http://www.seals-project.eu`

other reasoners perform consistency checking in a separate step after loading the ontology.

Ontology features - The evaluation needs to focus on a diverse set of publicly available ontologies which have different sizes (ranging from a few KB, to several MB), vocabulary sizes, structural characteristics and most importantly, different performance characteristics.

Reasoner benchmarking - The evaluation must perform classification on a number of ontologies using different publicly available reasoners. In this work, we will compare those reasoners that are actively-maintained, open-source and are able to support expressive languages such as OWL 2 DL.

Predictive models - The supervized machine learning technique, *classification*,[2] is used in the experiments to develop a predictive model to estimate inference time from metric values. Our goal is to be able to predict the membership of an ontology within a number of categories, defined over (discretized) reasoning time. A number of *classifiers* will be investigated to achieve the most effective prediction for different reasoners, since it is well-known that different classifiers will produce results of differing accuracies for different datasets.

For our specific problems of reasoner benchmarking and predictive model construction, we therefore first need to collect reasoning runtime data and metrics data. Secondly, we then need to leverage these metrics to develop a predictive model to determine the reasoning task time given the ontology metric values (for the subset of metrics that have the capacity to determine reasoning task time) and the reasoner. There are the following key steps in our approach:

1. *Data collection.* We need to collect a number of ontologies with a variety of characteristics, which may include recency, the application domain, file size, metric values, underlying ontology language, and most importantly, reasoning time. We also need to compute, for each ontology collected, its metric values, and an average time for the task of ontology classification. The *classification* reasoning task is performed on each ontology and the average reasoning time is recorded.

   Furthermore, since our goal is to learn predictive classifiers, we also need to discretize the continuous reasoning time in order to assign ontologies into separate groups based on their reasoning time.
2. *Building the predictive model.* The third stage of our approach constructs classifiers that classify ontologies into categories based on discretized reasoning time. The classifier typically builds a predictive model in the form of a Bayesian model, a decision tree, a regression model, or a set of rules. The prediction model is then evaluated for accuracy based on the widely-used 10-fold cross-validation. In this validation practice, each dataset is partitioned into k subsets. Each time, one of the k subsets is used as testing data, and the remaining (k-1) subsets form training data. The cross-validation process

[2] Note that this is an entirely different concept than ontology classification.

is then repeated k times with each of the k subsets used exactly once as the testing data. All k results from the folds can then be used as performance statistics. We use $k = 10$ as 10 is very often used in such validation practice.

4 Experiments and Analysis

Reasoners and reasoning task. We select four widely-used, actively-maintained and open-source reasoners that support OWL 2 DL, namely FaCT++ [26], HermiT [19], Pellet [23] and TrOWL [25] for our analysis of classification time. Note that TrOWL is incomplete because of the approximation it applies. In our experiment, CEL [2] is the underlying reasoner that TrOWL uses. The other three reasoners are complete OWL 2 DL reasoners. Table 1 below provides a brief summary of these reasoners.

Table 1. A brief summary of the four reasoners benchmarked.

	FaCT++	**HermiT**	**Pellet**	**TrOWL**
Version	1.5.3	1.3.5	2.3.0	0.8
Expressivity	OWL 2 DL	OWL 2 DL	OWL 2 DL	OWL 2 DL (partial)
Reasoning algorithm	Tableaux	Hypertableaux	Tableaux	Completion rules (CEL)

Consistency checking, another TBox reasoning task, is not selected. We observe that for some reasoners, consistency checking takes very short time on average (0.29s for HermiT and 0.05s for Pellet). At the same time, there is a very large discrepancy in consistency checking time between the four reasoners (mean: 4.02s for FaCT++ and 131.7s for TrOWL). Such a difference may be attributed to the different ways the reasoners report consistency checking (with or after ontology loading). Moreover, HermiT, Pellet and TrOWL all have a relatively normal distribution of consistency checking time. On the other hand, FaCT++ has quite a skewed distribution, where a single ontology takes more than 1,020 seconds while no other ontology takes more than 15 seconds. Hence, we believe it is not a fair comparison and we cannot draw useful conclusions from it.

The dataset. 358 real-world ontologies are collected, a large proportion of which are collected from the Manchester Tones Ontology Repository and NCBO BioPortal.[3] These ontologies vary significantly in file size, ranging from less than 4KB to almost 300MB. All ontologies collected from BioPortal are large ontologies with at least 10,000 terms. The expressivity of these ontologies spans simpler languages such as OWL 2 EL and QL, through OWL DL to OWL 2 DL and OWL Full, with a large number being in OWL 2 DL. At the same time, this collection also includes some well-known hard ontologies such as DOLCE, FMA, Galen, the Gene Ontology, the NCI Thesaurus and the Cell Cycle Ontology.

[3] `http://owl.cs.manchester.ac.uk/repository/` and `http://www.bioontology.org/`

Metrics As stated previously, we are interested in studying ontology metrics [28] and their capability in predicting classification time. Based on the metrics defined in [28], we propose a set of 27 metrics that we believe can characterize the structure and complexity of a given ontology. This set of metrics are derived from asserted logical axioms in an ontology are are divided into the following four categories:

- **Ontology-level (ONT)** metrics measure the size and structural characteristics of an ontology as a whole. Four ONT metrics are defined in [28]: SOV (size of vocabulary), ENR (edge node ratio), TIP (tree impurity) and EOG (entropy of graph). We define two additional metrics: CYC, that measures the Cyclomatic complexity of the ontology graph, and RCH, that measures the ratio between the number of anonymous class expressions and the total number of class expressions.
- **Class-level (CLS)** metrics measure the characteristics of OWL classes, which are first-class citizens in an ontology. Four such metrics are defined in [28], including NOC (number of children), DIT (depth of inheritance), CID (in-degree) and COD (out-degree).
- **Anonymous class expressions (ACE)** metrics count the total occurrences of each kind of anonymous class expressions that are available in OWL 2 DL. There are altogether 9 metrics: enumeration ($ENUM$), negation (NEG), conjunction ($CONJ$), disjunction ($DISJ$), universal/existential quantification (UF/EF) and min/max/exact cardinality ($MNCAR/MXCAR/CAR$).
- **Properties (PRO)** metrics measure the total occurrences of each kind of property declarations/axioms. The 8 PRO metrics records the number of occurrences of property declarations and axioms. There are 8 metrics, one each for: object/datatype property declaration (OBP/DTP), functional (FUN), symmetric (SYM), transitive (TRN), inverse functional ($IFUN$), property equivalence (EQV) and inverse (INV).

Data collection. For each ontology, values for the 27 metrics are collected. For each ontology and each reasoner, CPU time on ontology classification (but not loading) is averaged over 10 independent runs and recorded. All the experiments are performed on a high-performance server running OS Linux 2.6.18 and Java 1.6 on an Intel (R) Xeon X7560 CPU at 2.27GHz with a maximum of 40GB allocated (to accommodate potential memory leaks) to the 4 reasoner. OWL API [12] (version 3.2.4) is used to communicate with all four reasoners. Some hard ontologies take an extremely long time to classify. Hence, we apply a 50,000-second cutoff for all the reasoners.

4.1 Reasoner Performance Characteristics

The distributions of the raw reasoning time for the four reasoners can be found in Figure 1, where reasoning time is plotted in log scale due to its wide range (0s $\leq R \leq$ 50,000s), against ontologies sorted by their reasoning time. Note that

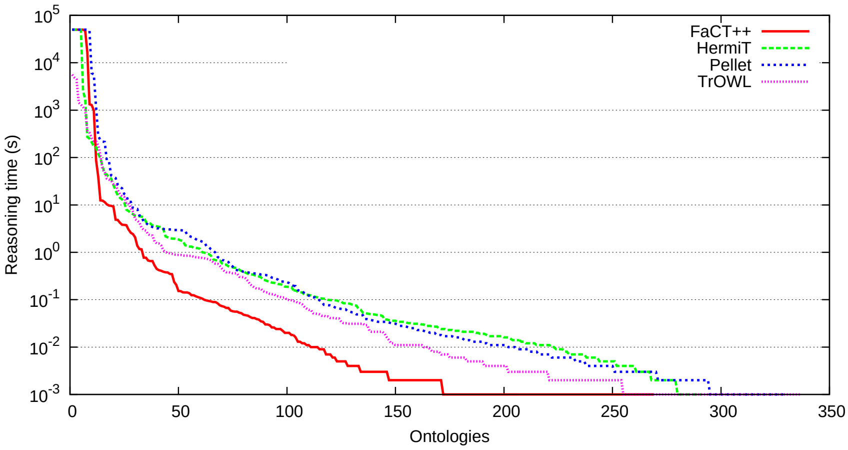

Fig. 1. Raw classification time of the four reasoners.

all reasoners except TrOWL time out (50,000 seconds) on a number of large and complex ontologies. As can be seen in the figure below, the distributions are highly skewed for all four reasoners.

Table 2 below provides some more details about the classification performance of the four reasoners, with the lowest value for each measure in **boldface** and the highest in *italic*. It can be seen in the second row that each reasoner fails to perform classification on a number of ontologies due to parsing or processing errors or the ontology being inconsistent.

It can be seen that for each reasoner, its mean is much higher than the median, indicating that the distribution is heavily skewed towards the right and that it may be the result of a small number of large values, which can be seen quite easily in Figure 1. It should also be noted that having a mean much larger than the median suggests that a distribution may be quite steep. This observation is confirmed by the high values of the skewness (Equation 1) and Kurtosis (Equation 2) measures in the table, which measure the (lack of) symmetry and the peakedness, respectively (s is the standard deviation of the sample). A skewness close to zero indicates roughly evenly distributed values. A positive skewness value indicates that the right-side tail of the distribution is longer than that of the left side, which is the case for all the four reasoners. A normal distribution has a Kurtosis measure of 0. A high Kurtosis value indicates that the data has a high peak, which is the case for all the four reasoners.

$$G_1 = \frac{n}{(n-1)(n-2)} \sum_{i=1}^{N} \left(\frac{x_i - \overline{x}}{s} \right)^2 \tag{1}$$

$$G_2 = \left(\frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^{N} \left(\frac{x_i - \overline{x}}{s} \right)^4 \right) - \frac{3(n-1)^2}{(n-2)(n-3)} \tag{2}$$

It can also be seen that the reasoners exhibit quite different performance characteristics. TrOWL and Pellet successfully complete on more ontologies than FaCT++ and HermiT. FaCT++, HermiT and Pellet all time out on a number

Table 2. Summary of raw classification time of the four reasoners.

	FaCT++	**HermiT**	**Pellet**	**TrOWL**
Number of ontologies resulted in error	*89*	67	28	**21**
Number of ontologies timed out ($> 50,000$s)	6	5	*8*	**0**
Mean (s)	1,366.4	879	*1,400.3*	**65.41**
Standard deviation (s)	7,967.41	6765.28	*8,121.11*	**490.07**
Median (s)	**0.002**	*0.037*	0.02	0.007
Skewness	**3.5**	7.46	5.81	9.56
Kurtosis	**10.63**	49.68	32	95.53

of ontologies, but not TrOWL. As a result of the clipping, the true performance value distributions for the former three reasoners may be even more skewed and peaked.

The performance of the four reasoners can be further characterized below.

FaCT++ has the lowest median, its distribution is the least skewed and also the least steep (lowest skewness and Kurtosis values) among the four. From Figure 1 it can be seen that FaCT++ performs the best on a large number of ontologies. However, it also fails on the most number (89) of ontologies (not due to clipping). **HermiT** has the highest median. However, its mean is the second lowest, after TrOWL. It also fails on quite many (67) ontologies. **Pellet** times out on the most number (8) of ontologies, indicating that Pellet may have trouble handling extremely large and difficult ontologies. Moreover, it has the highest mean and standard deviation, both of which are quite close to those of FaCT++. **TrOWL** has the lowest mean and standard deviation, both of which are much lower than those of the other three reasoners. This is due in part to the fact that TrOWL does not time out on any ontology. We note that TrOWL applies syntactic and semantic approximation, and hence is incomplete. Hence, the better performance may be the result of such incompleteness and requires further analysis. It is also noteworthy to point out that TrOWL has the most skewed and the steepest distribution among the four reasoners.

4.2 Predictive Model Construction

As stated previously, being able to predict reasoning performance using ontologies metrics is highly desirable for ontology engineering and ontology-enabled applications. In this work, we use *classification* in machine learning to build predictive models that accurately estimate reasoning performance of ontology classification. This section presents a major contribution of the paper. Namely, for each reasoner, we identify an accurate predictive model for reasoning time for the classification reasoning task, and a classifier used for determining the model.

As stated in the previous section, discretization is a necessary first step before classifiers can be trained. After raw run time values are collected, trivially simple ontologies (with reasoning time ≤ 0.01s) are removed from the dataset.

Experiments on the entire dataset without removal are also performed, where trained classifiers have even higher accuracy. However, this is due to the fact that the entire dataset is much more skewed towards simple ontologies. Hence the high accuracy is not really an improvement.

Reasoning time is then discretized into 4 bins uniformly, with unit interval width. The interval width is used as exponent of the reasoning time, i.e., 10^i is the cutoff point between bin i and bin $i+1$, for $1 \leq i \leq 4$. The bins are labelled 'A', 'B', 'C' and 'D'. A summary of the discretization and the size of the dataset for each reasoner in each bin is shown in Table 3.

Table 3. Discretization of reasoning time and number of ontologies in each bin.

Discretized label	Classification time	Fact++	HermiT	Pellet	TrOWL
A	A < 1s	75	154	126	105
B	1s ≤ B <10s	16	35	38	17
C	10s ≤ C < 100s	6	12	12	13
D	100s ≤ D	11	13	16	14
Total discretized		108	214	192	149
Trivial ontologies		161	77	138	188
Ontologies in error		89	67	28	21

It is well-known that reasoning performance is affected by the intrinsic characteristics of individual ontologies and that of the reasoner applied (underlying algorithms and optimization techniques). Hence, a single classifier may not be able to accurately model classification performance for all four reasoners. Hence, we employ a number of classifiers and identify the most effective one to build a predictive model for a given reasoner.

We use classification accuracy (simply accuracy) [18] to evaluate the effective of a classifier. Classification accuracy measures the percentage of correctly classified ontologies over all ontologies, and it is often considered to be the best performance indicator for evaluating classifiers in test data. As mentioned before, 10-fold cross validation is used to evaluate the classifiers and measure their performance in accuracy.

In total, we choose ten well-known classifiers available in Weka [27], with the aim of finding the best predictive models within an extensive spectrum. These classifiers are representative of six categories of classifiers: Bayesian classifiers (BayesNet and NaïveBayes), decision tree-based classifiers (J48 and RandomForest), rule-based classifiers (DecisionTable and OneR), a Support Vector Machine algorithm (SMO), a logistic regression-based classifier (SimpleLogistic), and instance-based classifiers (kNN, $1 \leq k \leq 10$, and K*).

Fig. 2 shows the accuracy of all ten classifiers for the four reasoners, using all 27 metrics as features. A number of important observation can be made.

- Three classifiers produce the best accuracy results for the four reasoners, with RandomForest performing the best for two reasoners, HermiT and Pellet.

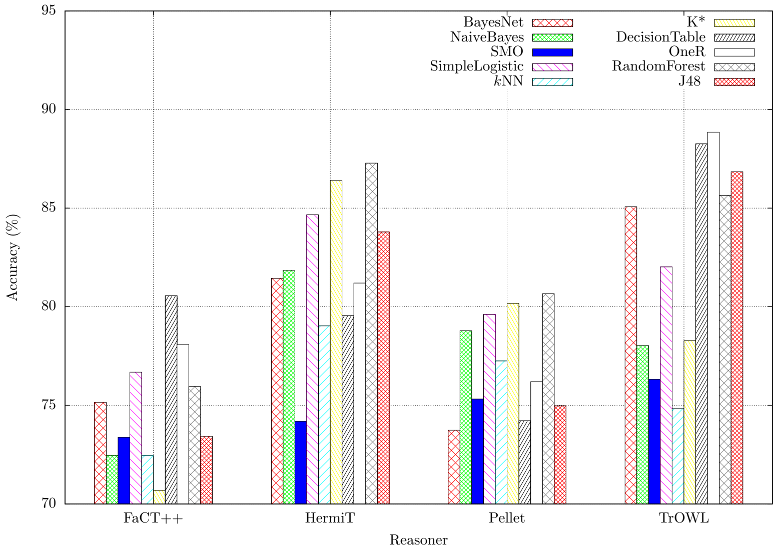

Fig. 2. Accuracy of all classifiers for the four reasoners.

- All the best classifiers for each reasoner achieve high accuracy, ranging from 80.56% for FaCT++ to 88.85% for TrOWL. Hence, it suggests that we can use such classifiers to predict classification performance with even higher accuracy.
- Overall, all classifiers produce consistently high accuracy, all higher than 70% with an average of 79.08%. This provides further evidence that (1) the predictive models found using our proposed approach can be effectively used for predicting classification performance; and (2) that ontology metrics can be used to learn predictive models for the classification task.

5 Conclusion

Our contributions in this paper are summarised are two-fold. Firstly, we study the classification performance of four widely-use, state-of-the-art OWL 2 DL reasoners, FaCT++, HermiT, Pellet and TrOWL (incomplete), comparatively. To the best of our knowledge, this is the most comprehensive of such studies in terms of the size and variability of the dataset (more than 300 ontologies with reasoning time ranging from subseconds to over 50,000 seconds). Some unique characteristics are discovered through our detailed study. Such charatceristics can be used in comparing and selecting reasoners for a given set of performance criteria.

Secondly, we further investigate the hardness of classification performance as a product of individual ontologies and reasoners. By applying machine learning techniques, we construct a model that can accurately predict performance with ontology metrics as features. Again, to the best of our knowledge, this is the first known study to apply machine learning techniques to predicting reasoning time for inference tasks. Experimental results confirm the effectiveness of our approach as the classifiers that are learned produce high (> 80%) accuracy for all the four reasoners.

Our future work will focus on further understanding the role individual metrics play in the predictive models and investigating their relative strength in predicting classification performance. We also plan to study a wider set of metrics in predicting reasoning performance. Though classification result is not the focus of this paper, we will compare that across reasoners to investigate their correctness. Moreover, we will investigate the feasibility of using metrics as a guide to generate synthetic ontologies that possess certain performance characteristics.

References

1. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope further. In K. Clark and P. F. Patel-Schneider, editors, *In Proceedings of the OWLED 2008 DC Workshop on OWL: Experiences and Directions*, 2008.
2. F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL—a polynomial-time reasoner for life science ontologies. In U. Furbach and N. Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR'06)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 287–291. Springer-Verlag, 2006.
3. F. Baader and W. Nutt. Basic description logics. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors, *The description logic handbook: theory, implementation, and applications*, pages 43–95. Cambridge University Press, 2003.
4. F. Baader and U. Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69(1):5–40, 2001.
5. S. Bail, B. Parsia, and U. Sattler. JustBench: a framework for OWL benchmarking. In *Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I*, ISWC'10, pages 32–47, Berlin, Heidelberg, 2010. Springer-Verlag.
6. P. Baumgartner, U. Furbach, and I. Niemelä. Hyper tableaux. In J. J. Alferes, L. M. Pereira, and E. Orlowska, editors, *JELIA*, volume 1126 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 1996.
7. J. Bock, P. Haase, Q. Ji, and R. Volz. Benchmarking OWL reasoners. In *ARea2008 - Workshop on Advancing Reasoning on the Web: Scalability and Commonsense*, June 2008.
8. D. Calvanese, G. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-Lite family. *J. Autom. Reason.*, 39:385–429, October 2007.
9. K. Dentler, R. Cornet, A. ten Teije, and N. de Keizer. Comparison of reasoners for large ontologies in the OWL 2 EL profile. *Semantic Web Journal*, 2(2):71–87, 2011.
10. T. Gardiner, D. Tsarkov, and I. Horrocks. Framework for an automated comparison of description logic reasoners. In *Proceedings of the 5th international conference on The Semantic Web*, ISWC'06, pages 654–667, Berlin, Heidelberg, 2006. Springer-Verlag.

11. B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 6:309–322, November 2008.
12. M. Horridge and S. Bechhofer. The OWL API: A java API for working with OWL 2 ontologies. In R. Hoekstra and P. F. Patel-Schneider, editors, *OWLED*, volume 529 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
13. I. Horrocks. Implementation and optimization techniques. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors, *Description Logic Handbook*, pages 306–346. Cambridge University Press, 2003.
14. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics*, 1(1):7–26, 2003.
15. I. Horrocks and U. Sattler. A tableaux decision procedure for $\mathcal{SHOIQ}$. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 448–453, 2005.
16. Y. Kazakov. Consequence-driven reasoning for horn $\mathcal{SHIQ}$ ontologies. In C. Boutilier, editor, *IJCAI*, pages 2040–2045, 2009.
17. M. Lawley and C. Bousquet. Fast classification in Protégé: Snorocket as an OWL 2 EL reasoner. In *Australasian Ontology Workshop 2010 (AOW 2010): Advances in Ontologies*, pages 45–50, Adelaide, Australia, 2010. ACS.
18. T. Mitchell. *Machine Learning*. Mcgraw-Hill International, 1997.
19. B. Motik, R. Shearer, and I. Horrocks. Hypertableau Reasoning for Description Logics. *Journal of Artificial Intelligence Research*, 36:165–228, 2009.
20. Z. Pan. Benchmarking DL reasoners using realistic ontologies. In B. C. Grau, I. Horrocks, B. Parsia, and P. F. Patel-Schneider, editors, *OWLED*, volume 188 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005.
21. R. Shearer, B. Motik, and I. Horrocks. HermiT: A Highly-Efficient OWL Reasoner. In *Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2008)*, 2008.
22. E. Sirin and B. Parsia. Pellet: An OWL DL Reasoner. In R. M. Volker Haaslev, editor, *Proceedings of the International Workshop on Description Logics (DL2004)*, June 2004.
23. E. Sirin, B. Parsia, B. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51–53, June 2007.
24. L. Steller, S. Krishnaswamy, and M. M. Gaber. Enabling scalable semantic reasoning for mobile services. *Int. J. Semantic Web Inf. Syst.*, 5(2):91–116, 2009.
25. E. Thomas, J. Z. Pan, and Y. Ren. TrOWL: Tractable OWL 2 Reasoning Infrastructure. In L. Aroyo, G. Antoniou, E. Hyvönen, A. ten Teije, H. Stuckenschmidt, L. Cabral, and T. Tudorache, editors, *ESWC (2)*, volume 6089 of *Lecture Notes in Computer Science*, pages 431–435. Springer, 2010.
26. D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006)*, pages 292–297. Springer, 2006.
27. I. H. Witten and E. Frank. *Data mining: Practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, San Francisco, 2000.
28. H. Zhang, Y.-F. Li, and H. B. K. Tan. Measuring Design Complexity of Semantic Web Ontologies. *Journal of Systems and Software*, 83(5):803–814, 2010.