How Can Reasoner Performance of ABox Intensive Ontologies Be Predicted?

Isa Guclu[1], Carlos Bobed[2], Jeff Z. Pan[1], Martin J. Kollingbaum[1], and Yuan-Fang Li[3]

[1]University of Aberdeen, U.K.
[2]University of Zaragoza, Spain
[3]Monash University, Australia

Abstract. Reasoner performance prediction of ontologies in OWL 2 language has been studied so far from different dimensions. One key aspect of these studies has been the prediction of how much time a particular task for a given ontology will consume. Several approaches have adopted different machine learning techniques to predict time consumption of ontologies already. However, these studies focused on capturing general aspects of the ontologies (i.e., mainly the complexity of their TBoxes), while paying little attention to ABox intensive ontologies. To address this issue, in this paper, we propose to improve the representativeness of ontology metrics by developing new metrics which focus on the ABox features of ontologies. Our experiments show that the proposed metrics contribute to overall prediction accuracy for all ontologies in general without causing side-effects.

Keywords: Semantic Web, Ontology Reasoning, Prediction, Random Forests, Knowledge Graph, Practical Reasoning

1 Introduction

Semantic technologies have been utilized in various application domains for assisting knowledge management thus far, e.g., data management [13] and software engineering [17]. The worst case complexity 2NEXPTIME-complete [6] of OWL 2 DL, the most expressive profile of OWL 2, constitutes a bottleneck for performance critical environments. Empirical studies show that even the EL profile, with PTIME-complete complexity and less expressiveness, can become too time-consuming [4, 11]. To have a scalable environment for implementing semantic technologies, an accurate prediction of ontology time consumption which will guide us about the feasibility of ontology reasoning is needed.

There have been several studies regarding the performance prediction of ontologies. Kang et al. [10] investigated the *hardness category* (categories according to reasoning time) for reasoner-ontology pairs and used machine learning techniques to make a prediction. Using FaCT++ [25], HermiT [5], Pellet [23], and TrOWL [20, 24, 18, 16], they reached high accuracy in terms of hardness category, but not reasoning time.

In another study, Kang et al. [12] investigated regression techniques to predict reasoning time. They made experiments using reasoners FaCT++, HermiT, JFact, MORe [21], Pellet and TrOWL with *their syntactic metrics* as features. These metrics are generally effective when there is a balance between TBox axioms and ABox axioms. Our experiments show that accuracy of these metrics decreases as ABox axiom sizes increase. As ABox constitutes the data in an ontology [1, 8, 27], where TBox constitutes the schema, an approach that can capture the changes in the ABox in a more detailed way is needed to make accurate overall predictions. As observed by Bobed et al. [2], there is an interest in using semantic technologies in mobile devices. In such scenarios, TBox axioms are expected to be more static and the ABox axioms (data) tend to be more frequently changing which necessitates high accuracy in ABox performance prediction. In this paper, we aim to investigate what metrics could help further improve reasoner predictions of ABox intensive ontologies.

Our main contributions can be summarized as follows.

1. We propose an initial set of metrics which estimate the complexity of the TBox concepts and propagates it into the estimated complexity of the ABox.
2. We show that our proposed new metrics for representing the structure of ontologies from the ABox perspective indicate a good research path to improve the accuracy of predicting time consumption of ontology reasoning.

The rest of the paper is as follows. In Section 2, we present some related works to place our proposal. In Section 3, we define the metrics that we propose in our ongoing work. In Sections 4 and 5, we explain our experimental settings and the achieved results, respectively. Finally, in Section 6, we make some conclusions and draw some future work.

2 Related Work and Background

Ontology metrics, which are features of the ontology expressed numerically or categorically to represent the structure of an ontology, have been effectively utilised in analysing the complexity [28], energy consumption on mobile devices [7], cohesion [26], quality [3] and population task [15] of ontology reasoning.

Kang et al. [10] proposed a set of metrics in 2012 to classify raw reasoning times of ontologies into five large categories: [0s.–100ms.], (100ms.–1s.], (1s.–10s.], (10s.–100s.] and (100s.–∞). Despite the high accuracy of prediction, over an 80%, this approach does not provide actual reasoning time but time categories, which may become obsolete or meaningless according to needs of implementation.

In 2014, Kang et al. [12] extended their work and proposed a new set of metrics to predict actual reasoning time by developing regression models. They extended the previous 27 metrics [10, 28] and developed a set of 91 metrics that include 24 ontology-level (ONT) metrics, 15 class-level (CLS) metrics, 22 anonymous class expression (ACE) metrics, and 30 property definition and axiom (PRO) metrics.

While a high number of metrics are usually proposed by researchers, Sazonau et al. [22] proposed instead a local method which involved selecting a *suitable*, small subset of the ontology, and making extrapolation to predict total time consumption of ontology reasoning using the data coming from the processing of such small subset. To do so, they used *Principal Component Analysis* (PCA) [9]. In their experiments, Sazonau et al. [22] observed that 57 of the studied features can be replaced by just one or two features. Using a sample of size of a 10% of the ontology for reasoning, they argue that they reached good predictions with simple extrapolations. They list advantages of their method as: 1) more accurate performance predictions, 2) not relying on an ontology corpus, 3) not being biased by this corpus, and 4) being able to obtain information about reasoner's behaviour of linear/nonlinear predictability on the corpus. A remarkable contribution of this approach is that it saves researchers from the difficulty/risk of selecting an unbiased corpus [14], which is very difficult while checking the validity of the prediction model and accuracy of the prediction. However, making reasoning with the 10% of an ontology may not always be applicable especially when the ontology requires high reasoning times.

3 Our Approach

Our claim is that increasing the expressivity of ontology metrics directly helps increasing the accuracy of all the above studies, and enables new studies that target a more feasible implementation environment for semantic technologies.

Part of 91 metrics proposed by Kang et al. [12] are obtained by transforming an ontology into a graph which grasps the relationship between of ABox and TBox axioms. However, their approach calculates the effect of ABox axioms up to a certain extent. It is apparent that connected ABox axioms are more prone to cause more inferences than disconnected ABox axioms. These connections can trigger reasoning time enormously when they come along with a complex TBox. In our work, we have observed that the models trained with this set of 91 metrics begin to lose accuracy in predicting time consumption of ontologies as the ratio between the amount of ABox axioms and TBox axioms increases.

Thus, we propose to include the propagation of the complexity of the TBox into the ABox. To do so, we extend this set of metrics with our 15 *Class Complexity Assertions (CCA)* metrics, which can contribute to performance prediction of ontologies especially when we deal with ontologies which are ABox intensive (i.e., they exhibit a high ABox/TBox ratio). Experiment results and source codes are accessible[1].

3.1 Class Complexity Assertions Metrics

As above mentioned, to capture the interactions between the complexity of the different elements of the TBox and the individuals asserted in the ABox, we have

[1] `http://sid.cps.unizar.es/projects/OWL2Predictions/JIST16/`

developed an initial set of features which aim at propagating the complexity of each of the concept expressions in the ontology to the ABox, as well as improving the richness of the TBox metrics.

Thus, let be $N_{CE} = \{CE_i \mid CE_i \in O\}$ with CE any concept expression appearing in any of the logical axioms of the ontology O. For each CE, we estimate its complexity as follows:

$$comp(CE_i) = \frac{height(CE_i) + sigSize(CE_i) + const(CE_i)}{3}$$

with $height(CE_i)$ being the height of the expression as a parsing tree, $sigSize(CE_i)$ being the number of different atomic class names that appear in the expression, and $const(CE_i)$ begin the number of class constructors participating in the class expression.

With this estimation for each CE_i, we calculate the following metrics:

- *TBoxSize:* The count of TBox axioms obtained from OWLAPI.
- *ABoxSize:* The count of ABox axioms obtained from OWLAPI.
- *ABoxTBoxRatio:* The ratio of ABox axioms to TBox axioms.
- *TCCA:* Total amount of estimated complexity of the ontology O (i.e., the class expressions in N_{CE}).

$$TCCA = \sum_{CE_i \in N_{CE}} comp(CE_i)$$

- *AVG_CCA:* Mean estimated complexity of the class expressions in N_{CE}.

$$AVG\_CCA = \frac{TCCA}{|N_{CE}|}$$

- *MAX_CCA:* Maximum estimated complexity of the class expressions in N_{CE}.
- *MIN_CCA:* Minimum estimated complexity of the class expressions in N_{CE}.
- *STD_CCA:* Standard deviation of complexity of the class expressions in N_{CE}.
- *ENT_CCA:* Entropy of the complexity distribution of N_{CE}.

To propagate the complexity of each concept expression to the ABox, we use each of the class assertions as a witness of the complexity of a class expression within the ontology. Then, we aggregate such values to capture what we name the witnessed complexity of the ABox. So, let $Ind_{N_{CE_i}} = \{a \mid a \in Ind(O) \wedge CE_i(a) \in O\}$ the individuals that are explicitly asserted to belong to CE_i. Thus, we define:

- *TWCCA:* Total witnessed complexity of the ABox, which is calculated summing all the products of the estimated complexities of the concept expressions with their *witness individuals*.

$$TWCCA = \sum_{CE_i \in N_{CE}} comp(CE_i) * |Ind_{N_{CE_i}}|$$

– *AVG_WCCA:* Mean witnessed complexity of the ABox of the concept expressions in O.

$$AVG\_WCCA = \frac{TWCCA}{|N_{CE}|}$$

– *MAX_WCCA:* Maximum witnessed complexity of a concept expression in O.
– *MIN_WCCA:* Minimum witnessed complexity of a concept expression in O.
– *STD_WCCA:* Standard deviation of witnessed complexity of the concept expressions in O.
– *ENT_WCCA:* Entropy of the witnessed complexity distribution of the concept expressions in O.

Note that we apply a Laplace smoothing[2] to include also into the metrics the concept expressions which appear in the ontology but do not have any explicit individual assertion.

4 Experimental Setup

4.1 Evaluation Metrics

R^2, $MAPE$ and $RMSE$ are referred to decide whether our regression model is valid for describing the relation between our metrics and the predictions made by the model. The coefficient of determination (R^2) is a crucial output of regression analysis, indicating to what extent the dependent variable is predictable. For example, a value 0.91 for R^2 means that 91 % percent of the variance in Y is predictable from X. Let $y(t)$ be the observed value of y in second t, $\hat{y}(t)$ be the predicted value for y in second t, and $\bar{y}$ be the mean of the observed values, then:

$$R^2 = \frac{\sum_t (\hat{y}(t) - \bar{y})^2}{\sum_t (y(t) - \bar{y})^2} \tag{1}$$

The *Mean Absolute Percentage Error (MAPE)* is a measure of prediction accuracy of a prediction method in statistics that is used to expresses accuracy as a percentage. For calculating the $MAPE$ of our prediction model, we will divide the difference of observed and predicted values, divide this by the observed values, and get the average of all observations in the scope. Related to this definition, we define the *Mean Absolute Accuracy Percentage (MAAP)* of our prediction model which is given by (1 - $MAPE$). In this paper, we will refer to $MAAP$ to explain the accuracy of a model.

$$MAPE = 100.\frac{\sum\limits_{t=1}^{n} \frac{|\hat{y}(t)-y(t)|}{y(t)}}{n} \tag{2}$$

$$MAAP = 1 - MAPE \tag{3}$$

[2] Adapted from Natural Language Processing, basically, it consists in adding 1 to all the witnessed values of the concept expressions in the ontology.

Finally, the Root Mean Squared Error ($RMSE$) is the square root of the mean/average of the square of all of the error. $RMSE$ represents the sample standard deviation of the differences between observed and predicted values.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (y(t) - \hat{y})^2}{n}} \tag{4}$$

4.2 Data Collection

Reasoner: We have used TrOWL 1.5 for testing EL ontologies as the reasoner to be tested. We deployed *ABox Materialization Task* with TrOWL as our experimental task. In our experiments, we implemented ABox materialization with one thread. We could benefit from parallelization in ABox materialization and it would improve the performance [19] to some extent. As RAM I/O becomes the bottleneck because of the limited bandwidth [19] of the RAM when many worker threads compete for RAM access and this would cause some side-effects in measuring the execution time, we preferred to analyse the performance prediction aspect parallel ABox materialization as future work.

Ontologies: We define an ontology as *ABox-intensive* if the count of ABox axioms in such an ontology is at least 10 times the count of TBox axioms. We made our experiments using ontologies in ORE2014 Reasoner Competition Dataset[3]. From 16,555 ontologies, we have filtered 74 ontologies in EL profile which have the ABox/TBox ratio of at least 10, and created artificial 2779 ABox-intensive ontologies[4] from these ABox-intensive ontologies *randomly* as follows: our method uses the TBox of the original ontology and creates a new ontology using different random subsets of the ABox axioms of the original ontology.

Prediction Model Construction: For predicting the time consumption of ontologies, a random forest based regression model is implemented, using the metrics (predictor variables). Standard 10-fold cross-validation is performed to ensure the generalizability of the model.

5 Results and Evaluation

In our study, we investigated the reasoning performance of a reasoner and ontology characteristics represented by available metrics and our new metrics (CCA). While developing our new metrics, we aimed at capturing the complexity of ontologies without losing accuracy when ABox sizes changed. Our claim is that developing high-quality metrics will increase the accuracy of the prediction model. Our goal is to make prediction models that can perform on any ontology with

[3] `https://zenodo.org/record/10791`

[4] You can find the code of the OntologyChopper at `http://sid.cps.unizar.es/projects/OWL2Predictions/JIST16/`

high stability using metrics that can represent the ontology with high expressivity.

To ensure the quality of the dataset, we created 2779 artificial ontologies from ORE2014 dataset. To avoid a biased corpus, which would result in misleading generalizations, we generated ontologies with random selection of ABox axioms. We did not put any threshold to cut the experiment, as we wanted to include every result of the dataset without missing any point that could be expressed by the dataset. We believe that wide range of ABox/TBox ratio will help increase the diversity in ontologies.

While working with the quality of the dataset, quality of the feature selection should also be taken into consideration. Inspired from the consistent high accuracy of the Random Forest based regression models in the study of Kang et al. [12], we adopted the same approach. Instead of categorising the time periods, we preferred metrics to give prediction results of time consumption in nanoseconds. We had specified R^2, $MAPE$, and $RMSE$ values as our performance criteria for prediction accuracy.

5.1 Combining 91 Metrics with CCA

In our first set of experiments, we combined 91 metrics with CCA metrics to train the model. We were expecting new CCA metrics would increase the accuracy of prediction, as it contained metrics that would better express the complexity of ABox axioms with TBox axioms. The results obtained in the cross validation procedure for the performance criteria can be seen in Table 1, and in Figure 1, the $MAPE$ values obtained are visualized.

When we look at the R^2 values, which is indicative to which extent the dependent variable is predictable, we see that both available 91 metrics and combined metrics of 91 metrics and CCA metrics have the values between 97% and 98%. The difference of $RMSE$ values is $\approx$ 2.5 seconds. The values of $MAPE$ also show a difference of $\approx$ 2%.

Although this absolute value of a $\approx$ 2 % accuracy increase seems very small, it is a relative improvement of 11 % with the first version of our transference metrics, which encourages us to continue to work in this direction improving and extending the definition of such kind of metrics.

	91 Metrics	CCA + 91 Metrics
R^2	0.97607	0.97856
$MAPE$	23.58 %	21.10 %
$RMSE$	41.4 sec.	39.1 sec.

Table 1: Contribution of CCA metrics to accuracy of prediction.

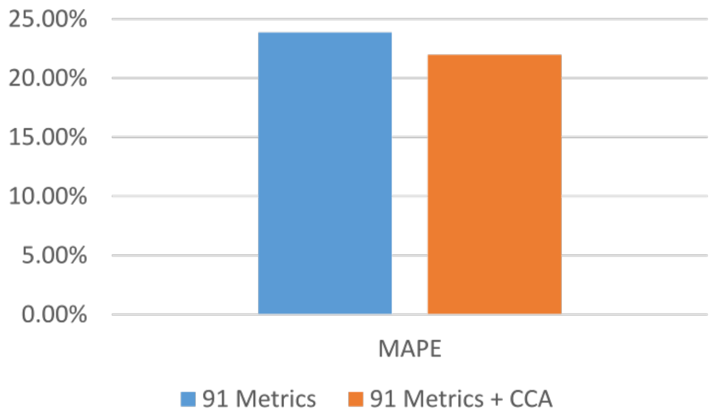

Fig. 1. Change in MAPE when new metrics are added to the prediction model.

5.2 Using CCA Metrics instead of *some* ABox metrics in 91 Metrics v.1

We searched for the metrics in 91 metrics which are more sensitive to ABox axiom changes. By randomly adding ABox axioms to ontologies, we observed that the change in ABox axioms is highly correlated with some of 91 metrics, i.e., "SOV, CYC, RHLC, IHR, IIR, ITR, IND, aCID, mCID, tCID" [10, 28].

In our second set of experiments, we removed the metrics "RHLC, IHR, IIR, ITR, IND, aCID, mCID, tCID" from 91 metrics and replaced with CCA metrics to train the model. The results obtained in the cross validation procedure for the performance criteria can be seen in Table 2, and in Figure 2, the $MAPE$ values obtained are visualized.

When we look at the R^2 values, we see that both models have the values between 97% and 98%. The difference of $RMSE$ values is $\approx$ 1 second. The values of $MAPE$ show a difference of $\approx$ 4.5%.

The relative improvement in decreasing the average error rate of 91 metrics is about 20 % by replacing some of ABox related metrics in 91 metrics with our CCA metrics.

	91 Metrics	CCA + 91 Metrics (v.1)
R^2	0.97607	0.97654
$MAPE$	23.58 %	19.03 %
$RMSE$	41.4 sec.	41.0 sec.

Table 2: Contribution of CCA metrics when replaced with some ABox related metrics in 91 metrics (v.1).

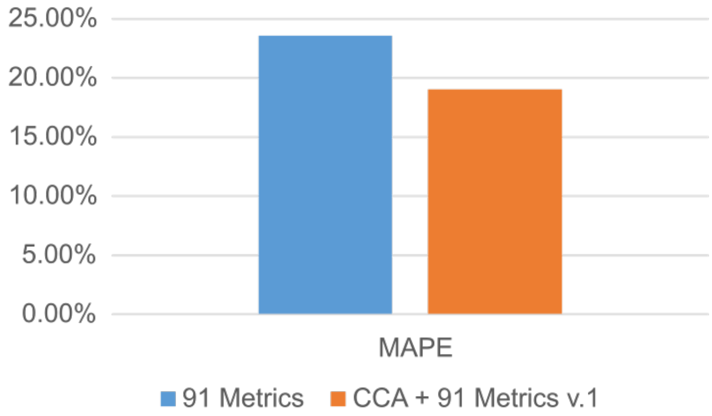

Fig. 2. Change in MAPE when some ABox related metrics in 91 metrics are replaced with CCA metrics (v.1) to the prediction model.

5.3 Using CCA Metrics instead of *some* ABox metrics in 91 Metrics v.2

In our third case, we removed the metric “CYC” in addition to “RHLC, IHR, IIR, ITR, IND, aCID, mCID, tCID” from 91 metrics and replaced with CCA metrics to train the model. The results obtained in the cross validation procedure for the performance criteria can be seen in Table 3, and in Figure 3, the $MAPE$ values obtained are visualized.

When we look at the R^2 values, we see that both models have the values between 97% and 98%. The value of $RMSE$ worsened here. The values of $MAPE$ show a difference of $\approx$ 4%, which is a general improvement but worst than the previous case.

The relative improvement in decreasing the average error rate of 91 metrics is about 18 % by replacing some of ABox related metrics in 91 metrics with our CCA metrics.

	91 Metrics	CCA + 91 Metrics (v.2)
R^2	0.97607	0.97449
$MAPE$	23.58 %	19.25 %
$RMSE$	41.4 sec.	42.7 sec.

Table 3: Contribution of CCA metrics when replaced with some ABox related metrics in 91 metrics (v.2).

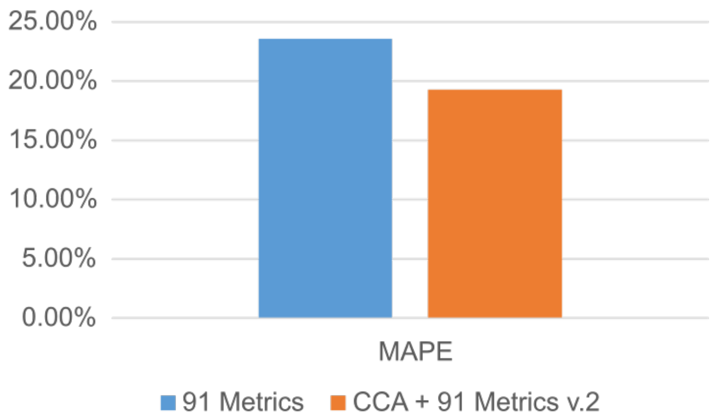

Fig. 3. Change in MAPE when some ABox related metrics in 91 metrics are replaced with CCA metrics (v.2) to the prediction model.

5.4 Evaluation

In our work, we have analysed available metrics and investigated how to bring expressivity of metrics further by developing new metrics to represent ABox axioms (and its interaction with TBox axioms) aspect of ontologies.

According to initial experiments, which compare 91 metrics with combination of 91 metrics and CCA metrics, we observe that adding CCA metrics increases the accuracy of prediction $\approx 2.5\%$ and relatively decreasing average error rate $\approx 11\%$.

When we replaced some of the metrics (RHLC, IHR, IIR, ITR, IND, aCID, mCID, tCID) in 91 metrics with CCA metrics, we observe the accuracy of prediction increase $\approx 4.5\%$ and relatively, average error rate decrease $\approx 20\%$.

In our third case, we also removed the metric (CYC) in 91 metrics and saw that there was again higher accuracy in prediction but it wasn't as good as the previous model.

Seeing the results above, we conclude that CCA metrics contributes to prediction of ABox-intensive ontologies in our preliminary work. Available metrics (91 metrics proposed by Kang et al. [12]) could grasp the complexity of ontologies to some extent. ABox materialization necessitates new metrics that will represent the interaction of ABox axioms with TBox axioms taking its complexity into account. The weight of ABox axioms in an ontology and their interactions can cause consuming more execution time than expected if their complexity is ignored. We propose our CCA metrics to measure the effect of ABox complexity in performance prediction of ontology reasoning and we want to improve these metrics to measure this aspect of ontologies more effectively. We believe that our study will lead to metrics that are generalizable regardless of the weight of TBox and ABox axioms.

6 Conclusion

Performance prediction of ontology reasoning is a very interesting and challenging topic. In this work, we have started to focus on the performance prediction of ABox-intensive ontologies. We proposed 15 new metrics by extending previous work of Kang et al. [12]. Preliminary results with adding these 15 metrics show slight increase (≈4.5%) in the prediction accuracy. And, these results even at the early stages of our research encourage us to continue in this direction. We believe that awareness of the ABox axiom ratio in ontologies and bringing a solution to this change will increase the effectiveness and validity of a performance prediction prediction model.

As future work, firstly, we plan to work on better representation of the interactions between ABox axioms and TBox axioms by developing new metrics. Secondly, we will make experiments with more reasoners on different ontologies that will help understanding the interaction of ABox axioms with TBox axioms in a broader sense. Thirdly, we will use different prediction mechanisms to leverage the contribution of these metrics.

Acknowledgments. This work was partially supported by the EC Marie Curie K-Drive project (286348), the CICYT project (TIN2013-46238-C4-4-R) and the DGA-FSE project.

References

1. Achille Fokoue, Felipe Meneguzzi, M. S., and Pan, J. Z. Querying Linked Ontological Data through Distributed Summarization. In *Proc. of the 26th AAAI Conference on Artificial Intelligence (AAAI2012)* (2011).
2. Bobed, C., Yus, R., Bobillo, F., and Mena, E. Semantic reasoning on mobile devices: Do androids dream of efficient reasoners? *Journal of Web Semantics, ISSN 1570-8268 35*, 4 (December 2015), 167–183. https://dx.doi.org/10.1016/j.websem.2015.09.002.
3. Burton-Jones, A., Storey, V. C., Sugumaran, V., and Ahluwalia, P. A semiotic metrics suite for assessing the quality of ontologies. *Data Knowl. Eng. 55* (2005), 84–102.
4. Dentler, K., Cornet, R., ten Teije, A., and de Keizer, N. Comparison of reasoners for large ontologies in the owl 2 el profile. *Semantic Web 2* (2011), 71–87.
5. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., and Wang, Z. Hermit: An owl 2 reasoner. *J. Autom. Reasoning 53* (2014), 245–269.
6. Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P. F., and Sattler, U. Owl 2: The next step for owl. *J. Web Sem. 6* (2008), 309–322.
7. Guclu, I., Li, Y.-F., Pan, J. Z., and Kollingbaum, M. J. Predicting energy consumption of ontology reasoning over mobile devices. In *15th Internaltion Semantic Web Conference (ISWC2016)* (2016).
8. Hogan, A., Pan, J. Z., Polleres, A., and Ren, Y. Scalable OWL 2 Reasoning for Linked Data. In *Reasoning Web. Semantic Technologies for the Web of Data. Lecture Notes in Computer Science 6848 Springer, ISBN 978-3-642-23031-8* (2011).

9. Jolliffe, I. Principal component analysis. *Wiley StatsRef: Statistics Reference Online* (2002).
10. Kang, Y.-B., Li, Y.-F., and Krishnaswamy, S. Predicting reasoning performance using ontology metrics. In *SEMWEB* (2012).
11. Kang, Y.-B., Li, Y.-F., and Krishnaswamy, S. A rigorous characterization of classification performance - a tale of four reasoners. In *ORE* (2012).
12. Kang, Y.-B., Pan, J. Z., Krishnaswamy, S., Sawangphol, W., and Li, Y.-F. How long will it take? accurate prediction of ontology reasoning performance. In *AAAI* (2014).
13. Li, Y.-F., Kennedy, G., Ngoran, F., Wu, P., and Hunter, J. An ontology-centric architecture for extensible scientific data management systems. *Future Generation Comp. Syst. 29* (2013), 641–653.
14. Matentzoglu, N., Bail, S., and Parsia, B. A corpus of owl dl ontologies. In *Proc. DL13* (2013).
15. Maynard, D., Peters, W., and Li, Y. Metrics for evaluation of ontology-based information extraction (2006).
16. Pan, J. Z., Ren, Y., and Zhao, Y. Tractable approximate deduction for OWL. In Arteficial Intelligence, volumn = 235, pages = 95-155.
17. Pan, J. Z., Staab, S., Amann, U., Ebert, J., and Zhao, Y. *Ontology-Driven Software Development.* Springer Publishing Company, Incorporated, 2012.
18. Pan, J. Z., Thomas, E., Ren, Y., and Taylor., S. Tractable Fuzzy and Crisp Reasoning in Ontology Applications. In *IEEE Computational Intelligence Magazine* (2012).
19. Ren, Y., Pan, J. Z., and Lee, K. Optimising parallel abox reasoning of el ontologies. In *Description Logics* (2012).
20. Ren, Y., Pan, J. Z., and Zhao, Y. Soundness preserving approximation for tbox reasoning. In *AAAI* (2010).
21. Romero, A. A., Grau, B. C., and Horrocks, I. More: Modular combination of owl reasoners for ontology classification. In *SEMWEB* (2012).
22. Sazonau, V., Sattler, U., and Brown, G. Predicting performance of OWL reasoners: Locally or globally? In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014* (2014).
23. Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. Pellet: A practical owl-dl reasoner. *J. Web Sem. 5* (2007), 51–53.
24. Thomas, E., Pan, J. Z., and Ren, Y. Trowl: Tractable owl 2 reasoning infrastructure. In *ESWS* (2010).
25. Tsarkov, D., and Horrocks, I. Fact++ description logic reasoner: System description. In *CADE* (2006).
26. Yao, H., Orme, A. M., and Etzkorn, L. Cohesion metrics for ontology design and application. *Journal of Computer Science 1* (2005), 107–113.
27. Yuan Ren, J. Z. P., and Lee, K. Optimising parallel abox reasoning of el ontologies. In *Proc. of the 25th Internatinoal Workshop on Description Logics (DL2012)* (2012).
28. Zhang, H., Li, Y.-F., and Tan, H. B. K. Measuring design complexity of semantic web ontologies. *Journal of Systems and Software 83* (2010), 803–814.