Efficient Reasoner Performance Prediction using Multi-label learning

Ashwin Makwana

Department of Computer Engineering, CSPIT, Charotar University of Science and Technology, CHARUSAT, Gujarat, India.

Abstract

The reasoner is the mechanism for interpreting the semantics of web ontology language. This paper focuses on reasoner performance study and predicting it by use of machine learning. Reasoner evaluation is very challenging as reasoner's efficiency may vary on different ontologies with the same complexity level. Different reasoners give different inference for the same ontology. Thus, reasoner could be enhanced for some however not for all ontologies. Here, paper focus on reasoner performance variability of reasoner and how ontology features affect reasoner performance. The main goal is to provide simple, efficiently computable guidelines to users. For prediction, supervised machine learning is used as a machine learning technique which help us to capture these dependencies. First introduced a new collection of efficiently computable ontology features, that characterize the design quality of an OWL ontology. Second, modeling of two learning problems: first, predicting the overall empirical hardness of OWL ontologies regarding a group of reasoners; and then, anticipating single reasoner robustness when inferring ontologies under some online usage constraints. To fulfill this goal, a generic learning framework is used, which integrates the introduced ontology features. The framework employs a rich set of machine learning models and feature selection methods. Furthermore, we used multi-label learning by analyzing the learned models unveiled a set of crucial ontology features likely to alter the empirical reasoner robustness.

Keywords

Semantic Web, Ontology, Reasoner, Multi-label learning, Supervised learning, Prediction.

Introduction

The problem that we want to focus is on semantic web reasoner performance measure and empirical assessment of multiple reasoners. An application developer can find the best suitable reasoner for given ontology. We proposed here machine learning techniques based on given ontology features to predict correctness or relevance and time for reasoning task by a set of reasoners. First, we have done experimentation for the empirical study of individual reasoner's performance prediction for its correctness and reasoning time using various ML model. After that, we proposed a multi-label classification technique to predict reasoning time and relevance of reasoner.

ISIC'21: International Semantic Intelligence Conference, February 25-27, 2021, Delhi, India.
EMAIL:ashwinmakwana.ce@charusat.ac.in (Ashwin Makwana)
ORCID: 0000-0002-4232-9598 (Ashwin Makwana)

Here, we have considered both profiles of ontology OWL (DL and EL). We have checked the performance parameter and compared it with the benchmark. The Semantic Web requires a standard, machine-processable representation of ontologies. The W3C has defined standard models and languages for this purpose. There are standard languages used for semantic web, Resource description framework (RDF) [1] and Web ontology language (OWL) [2]. Ontologies represented with these languages is becoming prevalent. These range from domain-specific ontologies, for example Gene Ontology. Semantic web Reasoner is one of the crucial components to fetch relevant knowledge from an ontology. To select appropriate reasoner is an essential task for a semantic web developer. During selecting reasoner, it has required to find a prediction of reasoner's performance.

Description logic-based Reasoners are crucial elements to work with OWL ontologies.

They are sued to produce explicit knowledge from ontologies to check their consistency and many other things. People are building an ontology by putting on domain knowledge and trying to get more expressive and representative ontologies. But more the ontology is expressive; the more reasoning is complex. In the worst case, reasoning can be non-deterministic doubled exponential. Thankfully, in practice, the reasoning is feasible even with very expressive ontologies. However, in general there is the theoretical complexity does not meet the empirical complexity.

There is an ample number of reasoners available for the semantic web application, and it is difficult for the application developer to choose right reasoner for an ontology for the domain-specific application. For evaluating ontology reasoners, OWL Reasoner Evaluation workshop organized every year. In this evaluation process, there are two significant issues one that is a disparity of reasoner's computing time which causes efficiency problem, i.e., for the same size and expressivity classes we get different computational time. The second problem is related to a disparity of reasoner's computed results, which produce correctness problem, i.e., for the same size and expressivity classes, we get different agreement level. For resolving above two issues, there are various explanations given by many researchers [3]–[6], but no tools available to cope with these phenomena.

Main research gap in this area is an exponential growth in a number of the reasoner; there is a variable empirical performance of reasoner. There is a lack of prior knowledge and expertise in this field. So main crux of this gap is how to help an application developer to choose the appropriate and suitable reasoner to work with domain-specific-ontology for a given application. To address these issues, many researchers [7]– [10] used machine learning techniques to learn reasoner's future behaviors from its past running for predicting single reasoner performances for, given an input ontology. Recently few works were carried out for predicting and ranking of a set of reasoners for, given an input ontology viz-a-viz, R2O2 [11] and RakSOR[12]. R2O2 is working on reasoning optimization technique, but there are issues in it [11], that it works only the runtime as criteria, there is no user assistance, there is the massive cost of the prediction steps and support DL ontologies only. While RakSOR[12] support user assistance and it takes runtime as well as correctness as criteria of ranking, but issues are; it uses a complicated and time-consuming process and only supports DL ontologies. Multi-RakSOR [13] uses the automatic ranking of ontology reasoners, which combines multi-label classification and multi-target regression techniques. It focused on the outcome as reasoner ranking and reasoner relevance prediction, which uses correctness and efficiency of reasoners as raking criteria. It also considers EL and DL both type of profiles of ontology. But it requires more optimization steps for improvement in performance. All these three-ranking works are not working on other reasoning tasks like consistency checking and realization, and not focused on memory and energy usage by reasoners.

Machine learning can bring a solution to this problem since it can help us to anticipate future reasoner behaviors by analyzing past running. Predicting single reasoners performance includes predicting the ranking of reasoners given an input ontology. However, all these solutions have many drawbacks. So, we proposed a new approach, the automatically rank reasoner, to recommend the fastest reasoner.

The contribution of this work is to proposed reasoner performance parameter prediction method. Which is experimented and implemented using ORE framework tool using Python libraries.

Literature Survey

Ontology Features and Metrics

Ontology features are qualitative and quantitative attributes covering structural and syntactic measures of a given ontology. Ontology metrics or functions are used for deciding reasoner performance prediction.

Use of ontology metrics can predict classification time. Based on parameters presented in [8], [14] proposed a twenty seven parameters that can be categorized the given ontology's complexity and structure. Many other metrices proposed in the literature [15] [16] [14] to measure various parameters of ontologies. In one article [15], authors estimate the quality of ontology like software engineering measures where, they shown a framework based on a suite of four metrics. Another group of authors [9] claim that metrics proposed by paper [7] are not sufficient, they used ML techniques and other ontology metrics for significant reduction in the dimensionality of various features of the ontology. Based on that, they identified vital features which correlated reasoners performance. Many numbers of ontology features were identified in the literature by the researcher for preparing learning models for reasoner classification time prediction. Recently one research group [10] reuse those feature and defied new features to compare to [9], they identify total 112 ontology features and split the ontology features into four categories like size description, expressivity description, structural features, and syntactic features. In one paper, authors [17] proposed a set of metrics which covers various points of ontology design. These metrics include all major ontology characteristics, and useful for reasoner performance prediction. Auto computation of these metrics is possible using efficient tools and methods, which help us to predict reasoners performance. In this paper [17] mainly eight ontology metrics were defined by considering ontology design-complexity. Ontology level and Class level are two main types of metrics. Each of the ontology, total twenty-seven distinct metrics are considered. **Figure *1*** shows the classification of various ontology metrics. These metrics are divided into categories like Ontology Size, Ontology Expressivity, Ontology Structure, and Ontology Syntax. These categories are further divided into various subcategories.

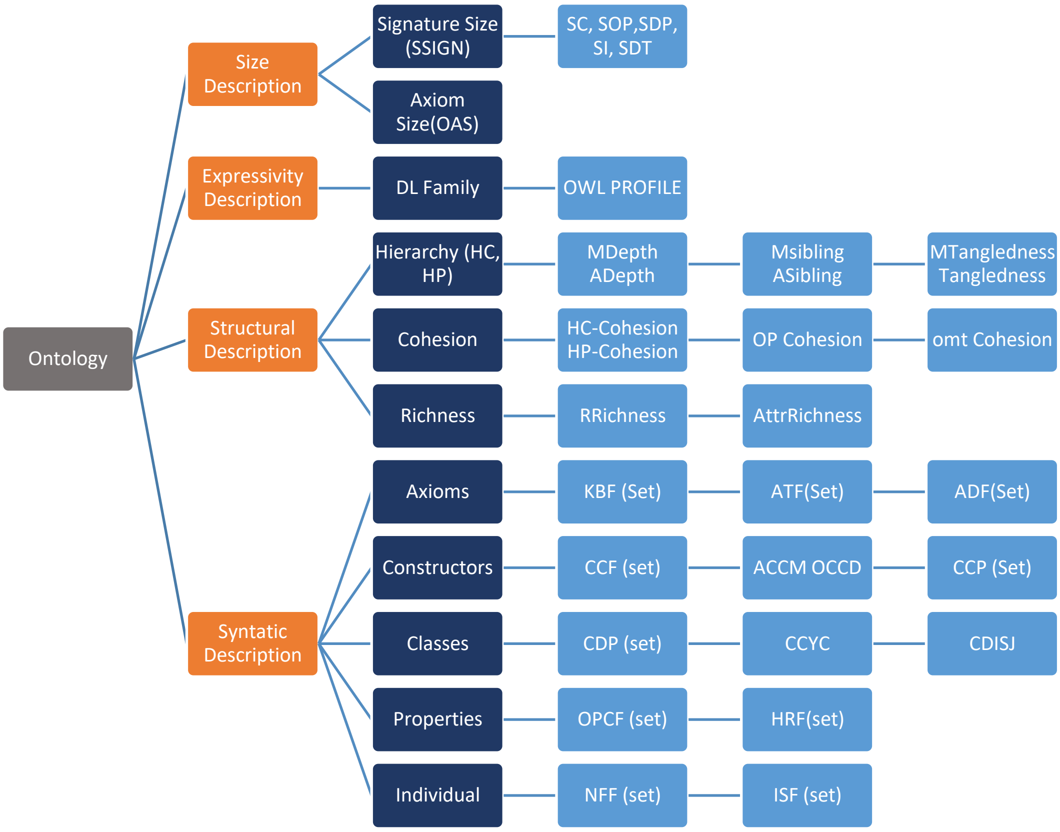

Figure 1: Ontology Features and Metrics classification

Survey on Reasoners

This brief study is to know the types of reasoner available with their characteristics and descriptions. Attributes of ontology reasoners. In paper [18] group of authors divide attributes of ontology reasoner in to 3 main category: *Reasoning characteristics*, *Practical usability and Performance indicators*. First, describes the basic features of ontology reasoners. Second, type of attributes determines whether the reasoner implements the OWL API. They also describe the availability and license of the reasoners. And last third, type is used to measure the performance of ontology reasoners. e.g., classification performance, TBOX consistency, checking performance, etc. There are various Reasoners available, comparative survey presented based on papers- [19], [20]. This survey covers ten major reasoners for the current study included in the scope of this paper.

Survey on Reasoners Performance Benchmark

There is a requirement to measure, benchmark, and characterize the performance of various reasoner available. The main aim of the SEALS project was to evaluate the DL-based reasoners. The comparison of three reasoners was made from standard inference servies. They have [8] used a data set of 300 ontologies and completed a comparative study which analyzes the performance of the reasoners. The reasoner performance for the ontology metrics by the usage of the machine learning techniques gave us a better idea about the complexity of the individual ontologies.
The classification is done on the ontologies using different reasoners. A comprehensive study is done regarding the variability and size of the dataset with more than three-hundred ontologies. They have also found some unique attributes with a thorough study. Such characteristics are used in reasoner's comparison and selection for given set of performance criteria.
Paper [21] focuses on benchmarking related to data sources and mappings to create more practical synthetic ontologies under managed conditions, we have used real-world ontology statistics to parameterize the benchmark. Workshop [22] focuses on bringing together both developers and users of reasoners for OWL comprising systems which can use the SEALS platform for their systems. Reasoning systems like jcel, FaCT++, WSReasoner, and HermiT were present. The OWL reasoner evaluation (ORE) [17] workshop encouraged the reasoned developers and ontology engineers to analyze the performance of new reasoners on OWL ontologies. The categorization, stability, and other factors for the reasoner were tested in the live and offline reasoned competition in the workshop. A total of 14 reasoners were submitted implementing specific subsets of OWL2. The reasoned competition is performed on many OWL ontologies obtained from the Web and the ontologies presented by the user.

Performance Prediction of Reasoner

Classification and Prediction are two main techniques of Machine Learning, especially in supervised learning, which required to apply during performance prediction of reasoners. Classification is ML technique which is used to identify the class for a new object like ontology, text or images, etc. from given set of classes.
Reasoner performance [8] is measured using various parameters. To judge performance parameter[9] before using in a semantic web application is a significant issue of research. Use of ontology metrics can predict classification time. Based on metrics presented in [14] [7] proposed total twenty-seven metrics of given ontology. Other proposed in the literature [15], [16], [23] to observe the quality, complexity, and cohesion of ontologies. Many numbers of ontology features were identified in the literature by the researcher for preparing learning models for reasoner classification time prediction. In recent literature by researchers in [10] reuse those feature and defied new features to compare to earlier work done in this area, they identify total 112 ontology features. We can use machine learning techniques to predict ontology classification performance.

Proposed Reasoner Prediction Framework

Semantic Web applications with ontologies, the behavior of reasoners used is very unpredictable. There are two main reasons for this; one reasoner would exhibit enormous scatter in computational runtime across the same ontologies and secondly, reasoners would derive different inferences for the same input ontology. These show the hardness of understanding reasoner's empirical behaviors for good reasoner developers.
For selecting the best reasoner for semantic web application using evaluation of reasoners performance, our hypothesis is that based on ontology features and metrics. We can predict reasoner's performance and can predict best-suited reasoner using machine learning techniques, especially using multi-label learning algorithms and ranking techniques.

Following are steps to follow for Reasoner Performance Prediction.

- Import Data contain ontology features and reasoner performance parameters. Data set of standard OAEI.
- Select standard Test data and Train data given in dataset.
- Define various features using feature selection, i.e., ontology characteristics and metrics. Define Target, i.e., Reasoning time, and Reasoner status.
- They fit multi-target classifiers for relevant and irrelevant reasoners for given ontology set.
- Arrange Reasoners for, given all ontology using relevant first and then according to the order of time after relevant reasoner put irrelevant reasoners according to the order of time.
- Give ranks according to the above arrangements.
- Fit classifier / Regression to predict ranks.

Reasoner Performance Prediction using ML

The main aim is to do work on automatically predict a reasoner's time efficiency and correctness. To achieve this goal, people have worked and suggested machine learning approaches, which includes the following steps. First, we required to work on the set of valuable ontology features, which will be used for learning ontology by machine learning model. ORE'2014 Framework is widely used to conduct experiments on various reasoners for their performance on given ontology corpus. At last, deployed supervised machine learning techniques to learn predictive models of reasoner performance based on previous execution. By interpreting these models, we can observer that few main features may change the performance of reasoner.

Feature selection is one of the prime steps in preprocessing dataset for training model in machine learning. The main purpose of feature selection is selecting the most relevant features by excluding non-useful features. Other researchers have used supervised discretization method (MDL), the Relief method (RLF); the CFS Subset method (CFS). We will have used Feature variance and Feature correlation with label data.

The supervised learning algorithms can be divided or grouped as logically based algorithms such as decision trees, Artificial Neural Network (ANN) based techniques such as multi-layered perceptron, Statistical learning algorithms such as Bayesian network and SVM. We can use some supervised machine learning algorithms like Random Forest, Simple Logistic Regression, Multilayer Perceptron, ANN-based learner, SMO SVM based learner and IBk K-Nearest-Neighbor based algorithm.

Multi-Label Learning for selection of Reasoner

Limitation of Single Label based learning is that it may not give consistent output for the selection of reasoner based on multiple criteria. Multi-Label based learning with multi-criteria is useful because single criteria may not give a consistent result.

The reason to apply multi-label classification is, for each ontology, there may be multiple possible correct reasoners. This inspired us to do multi-label classification for predicting relevancy of given ontology. Here Ranking of reasoners can be decided based on multiple criteria, i.e., like correctness or relevance of reasoner for given ontology and time taken for doing reasoning of that ontology. So, to decide out of all possible correct reasoners, we need to decide and identify the first one to experiment for given ontology. That is why we finalized two criteria for ranking reasoners that is correctness and time required for reasoning.

The solution of Reasoner selection methodology recently discussed and suggested by Alaya in [13] her paper on multi-label based learning for ontology reasoner's ranking. Based on this study author suggested that multi-label classification can be applied to reasoner based on ontology features. They have used Binary Relevance method [24] for Multi-label classification, which is one of the types of Problem Transformation method of MLC. For predicting they have used Multi-Target Regression, especially Ensemble of Regression chain[25].

In place of the above method to decide the better approach, we have done experiments with various ML model of MLC, where we found Ensemble Approach of MLC better compare to Binary relevance, especially we used Ensembled of Random Forest model. For MTR also, we have used Random forest for regression, which outperforms the Regression chain method.

If we compare the different problem transformation method for Multi-label learning, classifier chain is not advisable to exploit the correlation between targets. It gives a better chance of ranking higher to reasoner or label predicted last. Label power set method is not technically suitable for 1900 dataset in which a combination of 10 become more than 1000. In another word number of classes increase to more than 500, which is not good. Because of that, we have used Binary relevance as multi-label learning and Random forest as a base algorithm for multi-label learning because it performs better than KN, Logistic Regression, MLP, AdaBoost, Navi Bias, and QDA.

Assessment Measures

Evaluation and assessment measures are used to check the quality of ML model. For binary ML scenario, we could have TP, TN, FP, and FN value used for assessment. From this, we can calculate F1-measure, Precision, and Recall.

For assessment of ML with multi-class models, with an imbalanced dataset, we can use assessment measures like the F1-measure, Kappa coefficient, and Matthews the correlation coefficient. These measures we proposed to select the reasoner best predictive model. Assessment of relevance prediction model and compare with the existing system using Hamming loss and F1 measure.

Experimentation and Results

Experimentation Setup

Experiments to collect data for empirical behaviors of reasoners for classification task of a given set of OWL ontologies. For this we work with the evaluation tools in *ORE (Ontology Reasoner Evaluation Workshop)* [26] competition, which includes ORE Framework[1] and Ontology Corpora. We compare 10 reasoners for classification of 1900 distinct ontologies. For Reasoner Performance prediction, we have used Python language and *Jupyter Notebook* with python IDE. We have used Python library like *numpy, pandas, matplotlib, sklearn, xgboost, skmultilearn,* and their subclasses for prediction and classification of Reasoner performance.

Dataset

Ontologies data set is taken from ORE Corpora[2], around 1900 ontologies collected from this source which used for reasoner performance prediction.

Reasoners[3] set from popular categories are selected as candidates for performance evaluation and prediction process. Reasoner correctness/robustness and performance time is generated for 10 reasoners which have shown good efficiency in classification task of ontologies, during ORE competitions. The list of 10 reasoners includes *ELK, Konclude, MORe, ELepHant, HermiT, TrOWL, Pellet, FaCT++, Racer, JFact.*

Implementation

Start by evaluating the reasoners; we have to find empirical data. They describe the performance on a large set of ontologies. So, select to use tools proposed in the ontology reasoner evaluation workshop. We tool their framework ORE. We set classification challenges (DL & EL) 1900 ontologies. All the DL ontologies are to be handled by 8 reasoners, and 2nd challenge #EL ontologies will be handled by ten reasoners 8 + ELK and Elephant. A time limit of 3 minutes.

Steps for an experiment using Machine learning applied to estimate the best reasoner for ontology:

1. Import Data
2. Feature selection
3. Select test and train data
4. Apply ML methods for predicting reasoner relevance (for 10 reasoners)
5. Apply ML methods for predicting reasoner time (for 10 reasoners)

[1] ORE Framework-“https://github.com/andreas-steigmiller/ore-2014-competition-framework/”
[2] Ontology corpus - ”http://zenodo.org/record/10791”
[3] Reasoners - ”https://zenodo.org/record/11145”

6. Then select the best method for predicting relevance.
7. Predict reasoner's performance using multilabel classification/regression.

Result and Discussion

For Reasoner's performance prediction, we have applied various machine learning models like k-NN, Decision Tree, Random Forest, Neural Network, and AdaBoost. After applying this model on the dataset, we predicted Execution time as the target variable. We have measured and compare Error rate, i.e., Root Mean Square (RMS) Error given by each model for every 10 different reasoners. **Figure 2** shows that Random Forest is performing best for all ten reasoners compare to all other models. A neural network is the worst model for the majority of reasoner performance prediction.

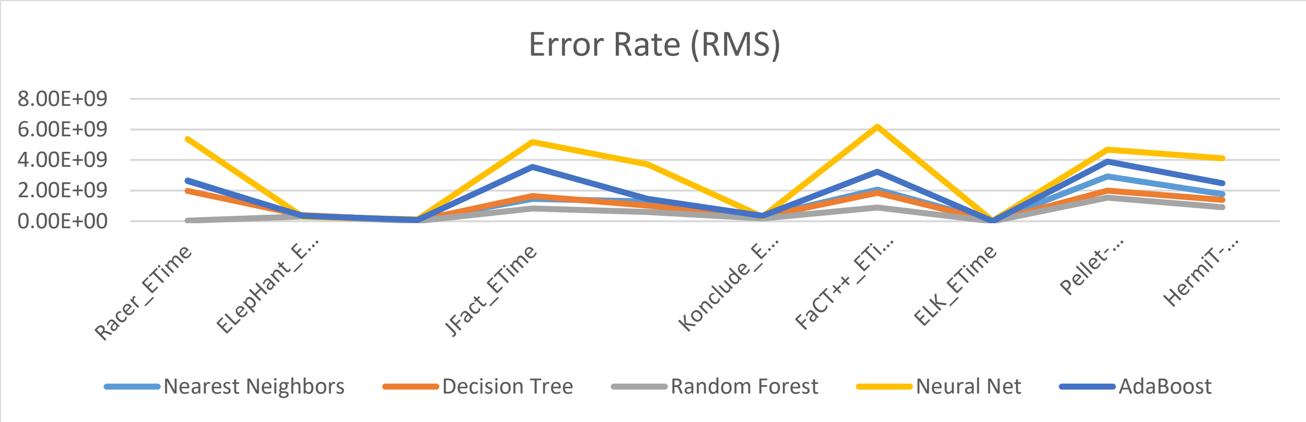

Figure 2 Prediction of Reasoner Execution Time using various ML

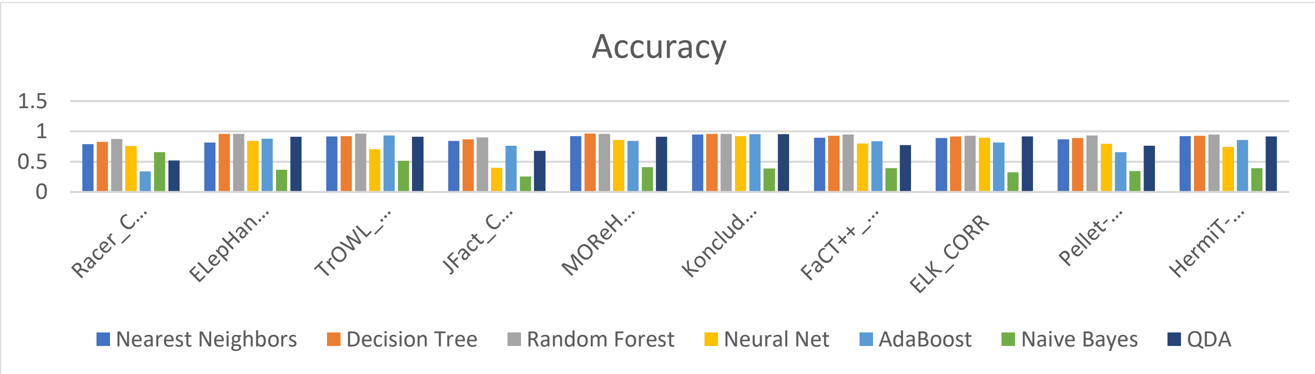

Figure 3 Accuracy of Relevance Prediction for all Reasoners using ML

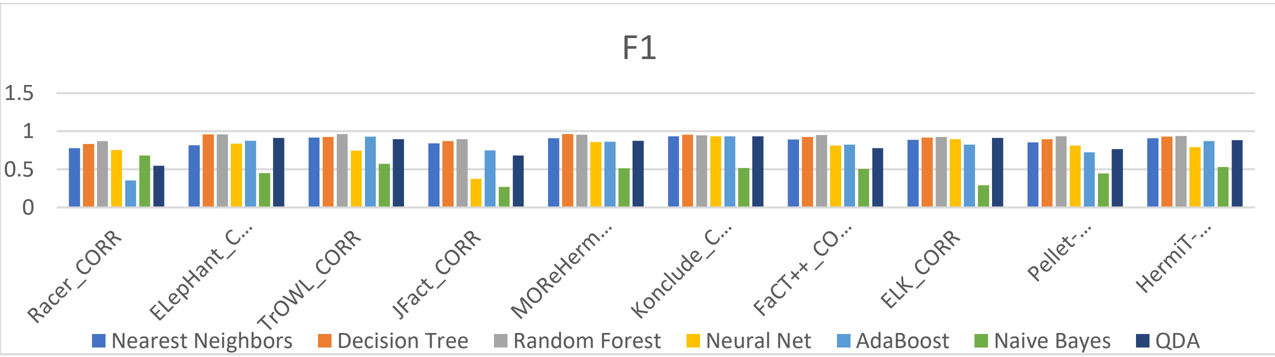

Figure 4 F1 measures of Relevance Prediction for all Reasoners using ML

Figure 3 shows a summary of all graph for accuracy Vs. Various reasoners for all ML models. We have also checked the performance parameter of prediction using F1 measure as per

Figure 4 graph. This graph exhibits that Radom Forest gives the best result in terms of F1 measure. By this graph, we can conclude the reasoner is the dominant reasoner in the DL ontologies. We can see that Hermit having a high rate of correctness is very slow, EL is dominant reasoner when in handling EL ontologies. All of this data will serve to create a learning data set. So, we try to divide the data in to Train and Test data to learn the mulitRAkSOR predictive models; then we assessed the relevance of the predictive quality of reasoner relevance. Our result shows our algorithm outperformed the existing solution.

Table 1 and **Figure 5** shows that Random forest shows significant improvement over other Multi-label learning models, including MulitRakSOR, especially for parameter Hamming-Loss and F1-measure.

We have used Multi-Label Classification using problem transformation methods and Adapted Algorithms like MLkNN, BPMLP, RAkEL, and Random forest. MLkNN It is a version of existing KNN for the multilabel learning task. It does not divide the problem into subproblems. BPMLP, this is a multi-label version of Neural Network-based algorithm. RAkEL is Random k Label set method. Random Forest special version for multi-label classification we have used.

We can observe results about multi-label learning method for prediction of reasoner's performance using the various parameters like Hamming-Loss, Accuracy, Jaccard-Similarity, Precision, Recall, and F1-measures.

Table 1 Multi-Label Learning Model Performance Analysis

	MLkNN	BPMLP	RAkEL	MultiRakSOR	Random Forest
Hamming-Loss	0.14	0.5	0.14	0.13	0.05
Accuracy	0.45	0	0.05	-	0.72
Jaccard-similarity	0.83	0.4	0.82	-	0.93
Precision	0.88	0.51	0.84	-	0.95
Recall	0.95	0.43	0.86	-	0.98
F1-Measure	0.91	0.4	0.85	0.95	0.97

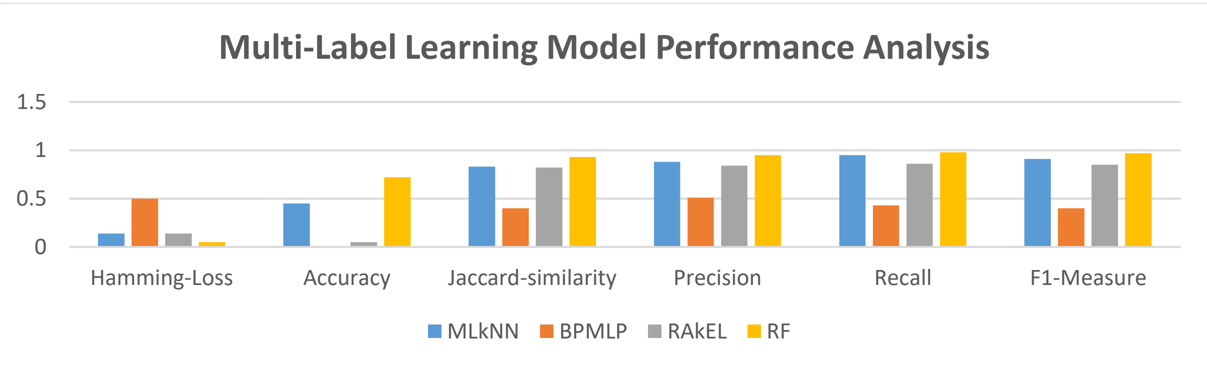

Figure 5 Multi-Label Learning Model Performance Analysis

Conclusions

For Semantic web heterogeneous store data in a structured way using Ontology concept, to fetch answer of the query of user, we required reasoner and logical rules. For understanding and using the semantic data on the web, there is a requirement of the reasoner.

For selecting appropriate reasoner by Semantic Web application developer, we have proposed a machine learning-based models for relevance and reasoning time prediction for given ontology. We have applied the multi-label learning method predicting the rank of various reasoners. Using single label prediction methods for given data set of ontologies and reasoner, we have shown that Random Forest is giving the best performance in terms of performance parameters. Same way, for multi-label learning model also Random forest variance outperform MLkNN, BPMLP, RAKEL and recently proposed MultiRakSOR in terms of Hamming-Loss (0.015), Accuracy, Jaccard-Similarity, Precision, Recall and F1-measures (0.97).

In future work, we could expand this approach for a greater number of ontologies and also on multiple domains. We could also extend our work in the future for SPARQL query performance measurement benchmark using a greater number of queries empirically by increasing number of experimentations.

References

[1] G. Klyne and J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax:," W3C Recommendation, Feb. 2004.

[2] G. Antoniou and F. van Harmelen, "Web Ontology Language: OWL," in Handbook on Ontologies, S. Staab and R. Studer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 67–92.

[3] T. Gardiner, D. Tsarkov, and I. Horrocks, "Framework for an Automated Comparison of Description Logic Reasoners," in The Semantic Web - ISWC 2006, Nov. 2006, pp. 654–667, doi: 10.1007/11926078_47.

[4] M. Lee, N. Matentzoglu, B. Parsia, and U. Sattler, "A Multi-reasoner, Justification-Based Approach to Reasoner Correctness," in The Semantic Web - ISWC 2015, vol. 9367, M. Arenas, O. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. Groth, M. Dumontier, J. Heflin, K. Thirunarayan, and S. Staab, Eds. Cham: Springer International Publishing, 2015, pp. 393–408.

[5] R. S. Gonçalves, B. Parsia, and U. Sattler, "Performance Heterogeneity and Approximate Reasoning in Description Logic Ontologies," in The Semantic Web – ISWC 2012, vol. 7649, P. Cudré-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J. X. Parreira, J. Hendler, G. Schreiber, A. Bernstein, and E. Blomqvist, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 82–98.

[6] T. D. Wang and B. Parsia, "Ontology Performance Profiling and Model Examination: First Steps," in The Semantic Web, vol. 4825, K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 595–608.

[7] Y.-B. Kang, Y.-F. Li, and S. Krishnaswamy, "Predicting Reasoning Performance Using Ontology Metrics," in The Semantic Web – ISWC 2012, vol. 7649, P. Cudré-Mauroux, J. Heflin, E. Sirin, T. Tudorache, J. Euzenat, M. Hauswirth, J. X. Parreira, J. Hendler, G. Schreiber, A. Bernstein, and E. Blomqvist, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 198–214.

[8] Y.-B. Kang, Y.-F. Li, and S. Krishnaswamy, "A Rigorous Characterization of Classification Performance - A Tale of Four Reasoners," 2012.

[9] V. Sazonau, U. Sattler, and G. Brown, "Predicting Performance of OWL Reasoners: Locally or Globally?," in Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning, Vienna, Austria, 2014, pp. 661–664, Accessed: Aug. 16, 2018. [Online]. Available: http://dl.acm.org/citation.cfm?id=3031929.3032020.

[10] N. Alaya, S. B. Yahia, and M. Lamolle, "Towards Unveiling the Ontology Key Features Altering Reasoner Performances," CoRR, vol. abs/1509.08717, 2015, Accessed: Aug. 16, 2018. [Online]. Available: http://arxiv.org/abs/1509.08717.

[11] Y.-B. Kang, S. Krishnaswamy, and Y.-F. Li, "R2O2:An Efficient Ranking-Based Reasoner for OWL Ontologies," in The Semantic Web - ISWC 2015, 2015, pp. 322–338.

[12] N. Alaya, S. Ben Yahia, and M. Lamolle, "RakSOR: Ranking of Ontology

Reasoners Based on Predicted Performances," in 2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), San Jose, CA, USA, Nov. 2016, pp. 1076–1083, doi: 10.1109/ICTAI.2016.0165.

[13] N. Alaya, M. Lamolle, S. Ben Yahia, N. Alaya, M. Lamolle, and S. Ben Yahia, "Multi-label Based Learning for Better Multi-criteria Ranking of Ontology Reasoners," in The Semantic Web – ISWC 2017, vol. 10587, F. Lecue, P. Cudré-Mauroux, J. Sequeda, C. Lange, and J. Heflin, Eds. Cham: Springer International Publishing, 2017, pp. 3–19.

[14] H. Zhang, Y.-F. Li, and H. B. K. Tan, "Measuring Design Complexity of Semantic Web Ontologies," J. Syst. Softw., vol. 83, no. 5, pp. 803–814, May 2010, doi: 10.1016/j.jss.2009.11.735.

[15] A. Burton-Jones, V. C. Storey, V. Sugumaran, and P. Ahluwalia, "A semiotic metrics suite for assessing the quality of ontologies," Data & Knowledge Engineering, vol. 55, no. 1, pp. 84–102, 2005,doi:https://doi.org/10.1016/j.datak.2004.11.010.

[16] H. Yao, A. M. Orme, and L. Etzkorn, "Cohesion metrics for ontology design and application," Journal of Computer science, vol. 1, no. 1, pp. 107–113, 2005.

[17] S. Bail et al., Eds., Proceedings of the 2nd International Workshop on OWL Reasoner Evaluation (ORE-2013), Ulm, Germany, July 22, 2013, vol. 1015. CEUR-WS.org, 2013.

[18] K. Dentler, R. Cornet, A. ten Teije, and N. de Keizer, "Comparison of Reasoners for Large Ontologies in the OWL 2 EL Profile," Semant. web, vol. 2, no. 2, pp. 71–87, Apr. 2011, doi: 10.3233/SW-2011-0034.

[19] A. Khamparia and B. Pandey, "Comprehensive Analysis of Semantic Web Reasoners and Tools: A Survey," Education and Information Technologies, vol. 22, no. 6, pp. 3121–3145, Nov. 2017, doi: 10.1007/s10639-017-9574-5.

[20] N. Matentzoglu, J. Leo, V. Hudhra, U. Sattler, and B. Parsia, "A Survey of Current, Stand-alone OWL Reasoners," Informal Proceedings of the 4th International Workshop on OWL Reasoner Evaluation (ORE-2015) co-located with the 28th International Workshop on Description Logics (DL 2015), Athens, Greece, June 6, 2015., pp. 68–79, 2015.

[21] Y. Li, Y. Yu, and J. Heflin, "Evaluating Reasoners Under Realistic Semantic Web Conditions.," 2012.

[22] I. Horrocks, M. Yatskevich, and E. Jiménez-Ruiz, Eds., Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012), Manchester, UK, July 1st, 2012, vol. 858. CEUR-WS.org, 2012.

[23] H. Zhang, Y.-F. Li, and H. B. K. Tan, "Measuring design complexity of semantic web ontologies," Journal of Systems and Software, vol. 83, no. 5, pp. 803–814, 2010, doi: https://doi.org/10.1016/j.jss.2009.11.735.

[24] M. Ioannou, G. Sakkas, G. Tsoumakas, and I. P. Vlahavas, "Obtaining Bipartitions from Score Vectors for Multi-Label Classification," in 22nd IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2010, Arras, France, 27-29 October 2010 - Volume 1, 2010, pp. 409–416, doi: 10.1109/ICTAI.2010.65.

[25] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, and I. Vlahavas, "Multi-target regression via input space expansion: treating targets as inputs," Mach Learn, vol. 104, no. 1, pp. 55–98, Jul. 2016, doi: 10.1007/s10994-016-5546-z.

[26] B. Parsia, N. Matentzoglu, R. S. Gonçalves, B. Glimm, and A. Steigmiller, "The OWL Reasoner Evaluation (ORE) 2015 Resources," in The Semantic Web – ISWC 2016, vol. 9982, P. Groth, E. Simperl, A. Gray, M. Sabou, M. Krötzsch, F. Lecue, F. Flöck, and Y. Gil, Eds. Cham: Springer International Publishing, 2016, pp. 159–167.