Large Scale Knowledge Base Systems: An Empirical Evaluation Perspective

Yuanbo Guo, Abir Qasem, and Jeff Heflin

Computer Science & Engineering Department, Lehigh University
Bethlehem, PA18015, USA
{yug2, abq2, heflin}@cse.lehigh.edu

Abstract

In this paper, we discuss how our work on evaluating Semantic Web knowledge base systems (KBSs) contributes to address some broader AI problems. First, we show how our approach provides a benchmarking solution to the Semantic Web, a new application area of AI. Second, we discuss how the approach is also beneficial in a more traditional AI context. We focus on issues such as scalability, performance tradeoffs, and the comparison of different classes of systems.

Benchmarking Semantic Web KBSs

Our research interest is to develop objective and unbiased ways to evaluate Semantic Web knowledge base systems (KBSs) (See Guo, Pan and Heflin 2004). Specifically, we have conducted research on benchmarking KBSs that store, reason and query statements described in OWL[1], which is a standard language for describing and publishing Web ontologies. As a product of our work, we have developed the Lehigh University Benchmark (LUBM). The LUBM is, to the best of our knowledge, the first of its kind and has become well recognized in the Semantic Web community. The LUBM is designed to fill a void that we consider particularly important, i.e., the evaluation of systems with respect to large instance data that commit to an ontology of moderate size. In creating the benchmark, we have developed:

1) An OWL ontology for the university domain.
2) A technique for synthetically generating instance data over that ontology. Importantly, this data can be regenerated given only a seed and can be scaled to an arbitrary size. Moreover, to make it as realistic as possible, the data is generated by obeying to a set of restrictions that are elicited from an investigation into the domain (e.g. the ratios between instances of different classes and the cardinality of different properties for individuals of different types).
3) Fourteen test queries against the instance data. These queries have been chosen to represent a variety of properties including input size, selectivity, complexity, and assumed logical inference.
4) A set of performance metrics including data loading time, repository size, query response time, and degree of query completeness and soundness. We have developed these metrics by borrowing from standard database benchmarks and at the same time trying to address the unique properties of the Semantic Web. Later we will have more discussion on this.

There are several benefits to our benchmarking approach. The benchmark facilitates the evaluation of systems with respect to two basic and conflicting requirements: first, the enormous amount of data means that scalability and efficiency become crucial issues; second, the system must provide sufficient reasoning capabilities to support the semantic requirements of a given application.

Another key benefit of our approach is that it allows us to empirically compare very different systems. For instance, we have conducted a benchmark experiment on the following KBSs: Sesame (Broekstra and Kampman 2002), DLDB-OWL (Pan and Heflin 2003), and OWLJessKB (Kopena and Regli 2003). These systems represent distinct points in terms of OWL reasoning support as well as reasoning mechanisms. Specifically, Sesame supports the RDFS[2] language and is incomplete with respect to OWL. Its reasoning is forward-chaining style. DLDB-OWL uses FaCT (Horrocks 1998), a description logic reasoner based on the tableaux algorithms, to precompute certain ontological information. However, for scalability considerations, DLDB-OWL translates queries into SQL and issues them to a relational database management system. OWLJessKB uses a production system as its underlying reasoner and among the systems supports the most OWL reasoning. Moreover, these systems differ in their storage mechanisms: OWLJessKB manipulates data in main memory while DLDB-OWL is based on persistent storage; also we have tested both the main memory-based and the database-based versions of Sesame (hereinafter we refer to them as Sesame-Memory and Sesame-DB respectively). The experimental results helped us characterize the

[1] Noticeably, OWL is based on Description Logics.

[2] In terms of expressivity, RDFS is similar to a Semantic Network.

performance of each system. In particular, we were able to show how these systems compare in terms of the performance tradeoff they make and the corresponding impact in large data situations. We will give some examples of this in the appropriate context later on.

The LUBM is a benchmark limited to a particular domain. Ideally we should have a suite of benchmarks representing different domains with different workloads. In light of this, we have extended our work with an approach for rapid development of such benchmarks (Wang et al. 2005). Given training data for a domain, the approach is able to learn a model that can be used to generate representative synthetic data. Specifically, the algorithm extracts certain statistical features of the training data and accordingly constructs a probabilistic model (e.g. the probability of an individual belonging to a specific class and the probability of an individual of a specific class having a specific cardinality for a specific property). Then based on the model, a Monte Carlo algorithm is used to generate synthetic data that has similar properties to the training data. This approach helps overcome the problem of having insufficient real world data for benchmarking and allows us to develop benchmarks for a variety of domains and applications in a very time efficient manner.

In the remainder of the paper, we will discuss how our work is beneficial to the AI community.

The Semantic Web as a New Test Bed for AI and Our Benchmarking Solution

The Semantic Web envisions a web of ontologies and associated data, which will augment the present Web with formal semantics. We can view the Semantic Web as a new AI problem that aims at representing knowledge in a huge, open and distributed environment. Thus existing AI research could serve as the starting point in solving the problem. At the same time, many issues will arise in applying and tailoring the traditional AI techniques and systems to the Semantic Web.

Frank van Harmelen (2002) has identified some assumptions underlying Knowledge Representation (KR) that need to be revised when applied to the Semantic Web and the associated challenges and issues. In summary, he has put forth the following issues: 1) Scale. Much larger knowledge bases than traditional KR systems are designed for, 2) Higher change rate of information and unpredictable use of knowledge, 3) Having to deal with the cases that portions of a knowledge base are missing, 4) Trust and justification. Statements in a knowledge base may be of different level of credibility as well as quality, 5) Multiple knowledge sources and diversity of content, 6) Need for remotely linking to knowledge bases and accessing their statements, 7) Robust inferencing with possible incompleteness and unsoundness.

Our work contributes to addressing some of the above issues from a benchmarking point of view. First, we have placed great emphasis on the evaluation of systems with respect to scalability. A key assumption of our work is that future Semantic Web systems will need to reason with massive instance data. In particular, we believe that instance data will by far outnumber ontologies. This trend is already beginning to emerge. According to SWOOGLE (Ding, L. et al. 2005), which has indexed over one million Semantic Web documents, the ratio of data documents to ontologies was about 80 to 1 in 2005, and this gap has widened by 40% in 2006.

Given that there are no KBS evaluation methods with this focus, we have developed a technique for generating instance data over the benchmark ontology and this data can be scaled to an arbitrary size. This allows us to gain an insight into a system's scalability by testing it against a range of sizes of data. For example, in the aforementioned experiment, we discovered that Sesame-Memory could load a larger size of data than we had expected. Furthermore, we were able to identify the data size beyond which the performance of Sesame-Memory would go down dramatically.

The second feature of our work is that we do not assume the system under test is complete and sound in reasoning. Recall that an inference procedure is complete if it can find a proof for any sentence that is entailed by the knowledge base. With respect to queries, we say a system is complete if it generates all answers that are entailed by the knowledge base, where each answer is a binding of the query variables that results in an entailed sentence. However, in an environment such as the Semantic Web, partial answers will often be acceptable. So it is important not to measure completeness with such a coarse distinction. Instead, we provide metrics for measuring the degree of completeness and soundness (abbreviated as *doc* and *dos* respectively), as follows.

entailed: *the set of entailed answers to q*
returned: the set of answers to q returned by the system
correct: *entailed* ∩ *returned*

$$doc = \frac{|correct|}{|entailed|} \qquad dos = \frac{|correct|}{|returned|}$$

Note, *doc* and *dos* are analogous to the standard metrics of recall and precision in Information Retrieval respectively.

Notice that the above two metrics are intended to complement, not replace, theoretical analysis. Techniques such as the alternative semantic accounts (based on weaker, 4-valued logics, for example) (Patel-Schneider 1989) and proof-theoretic semantics (Borgida 1992) can be used to characterize the incompleteness of systems. However, we would like to point out that just because a KBS is incomplete does not mean it will be incomplete for a specific application. In particular, we believe there will be much redundancy on the Semantic Web: there may be many different ways to derive facts and often the derivable facts will be stated explicitly elsewhere. Thus we see degree of completeness and degree of soundness as measures of a KBS's performance on a specific kind of workload (consisting of an ontology, data and queries).

Third, in line with the above discussion, we could expect a variety of reasoning capabilities and strategies in

Semantic Web KBSs. Our benchmark is designed to help the user measure potential tradeoff associated with those factors and answer questions such as if the speed/scalability gain is worth the sacrifice in reasoning/query completeness. For instance, in the experiment we have conducted, compared to OWLJessKB, Sesame and DLDB-OWL perform less complete reasoning, however, they could generally load data and answer queries faster than OWLJessKB. Also, Sesame carries out all the inferences during loading. This appears to make most queries faster, but results in greater repository size and load time and significantly limits the ability of the system to load large dataset sizes.

As a more concrete example, consider the examination of query response time and query completeness at the same time. The figure below demonstrates one of our recommended ways for doing that. In the chart, we use clustered columns to compare query response time, and in addition, the percentage of the filled area of each column to indicate the degree of query completeness. This kind of interpretation makes it easy to compare systems in terms of both metrics. For example, for a specific query, the user may decide that a system with a fast query time but an empty bar should not be favored over a system that is slower but with a more complete bar.

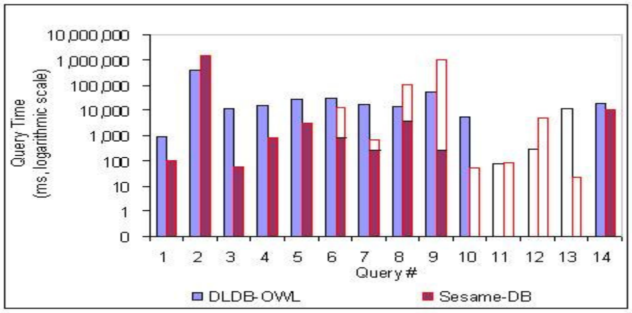

Fig. 1. Query Response Time with Query Completeness.

Applying Our Work to Traditional AI

As can be seen, although our work evaluates KBSs for OWL, our benchmarking approach is not tied to a specific language nor it is to the specific area of the Semantic Web. For example, the approach could apply when evaluating a first-order logic KBS, where the knowledge base consists of a set of axioms and ground facts (analogous to OWL ontology and instance data respectively). Similarly, the system under test could be one that uses KIF as its query language as opposed to a more typical Semantic Web query language. Also we can use a similar approach to evaluate an intelligent agent system that utilizes knowledge representation and reasoning techniques to model a complex domain with a semantically rich language.

At its core, our approach is an attempt to address some key questions that are commonly faced by the evaluators of any AI KBSs. The first question is how to empirically evaluate the system's scalability? Scalability can be a critical requirement when we are applying AI techniques in a large-scale intelligent system, or when we are deploying systems in an environment such as the Web. In order to evaluate the systems in terms of its scalability, we need datasets that are representative enough of the intended application and at the same time are of very large sizes. However, it may not be easy to acquire satisfactory test data from the real world due to that we are interested in an emerging field of application, which still lacks real world data. Or it could be because that, with the real world data available, it is hard to control specific factors such as query selectivity and reasoning requirements. Our work represents an alternative solution to this issue, that is, we can generate synthetic data that simulates the intended domain. Moreover, we are open to consider partial answers in favor of scalability. This pragmatic approach is what fundamentally distinguishes our work from some similar performance evaluation approaches in AI.

The second important question is how to empirically compare and select different KBSs? It is often the case that a user is faced with the choice of different classes of systems. One such example from the AI literature is when the authors try to compare a description logic reasoner versus a first-order logic prover (Tsarkov and Horrocks 2003). Related to the above question is how we evaluate systems with respect to conflicting requirements. For instance, increased reasoning capability usually means an increase in data processing time and/or query response time. Oftentimes, it is insufficient to only look at the theoretical properties of the systems. For example, although some logical reasoners are incomplete with respect to the language in question, they may still be useful because they scale better or respond to queries more quickly. We could further broaden this notion of tradeoff. For instance, in the area of intelligent agent, one has to continuously deal with resource-boundedness and make tradeoff decisions (Lesser et al. 2000). Specifically for example, while doing approximate reasoning, a tradeoff has to be made between decision quality and computational cost (Zilberstein 1995). We can view the optimal decision of an intelligent agent as the counterpart of sound and complete reasoning of KR.

In KR, traditionally it has been accepted that there is a tradeoff between scalability and reasoning completeness. As such, one of the most enduring challenges in KR is to push systems so that they are better in both properties. Consider Fig. 2, which we consider roughly represents a specific case which most people would find reasonable nowadays. A long-term research agenda for KBS development is to push the frontier outward. In this regard, our approach can help researchers and developers understand the-state-of-the-art and identify research issues and directions through empirical evaluation of the systems and we consider this is equally important as pure theoretical analysis.

Next, we discuss some work on evaluating KBSs in traditional AI literature. The automated theorem prover (ATP) competitions (Sutcliffe and Suttner 2004) evaluate the performance of sound, classical first order ATP

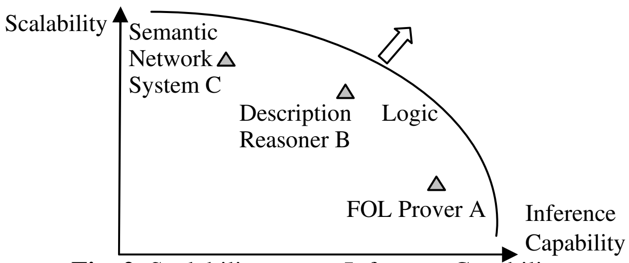

Fig. 2. Scalability versus Inference Capability

systems in terms of the number of problems solved and the average runtimes for those problems. Since their emphasis is on evaluating the reasoning algorithms, they discard systems that yield partial answers to the problem.

One of the earliest efforts to evaluate large KBSs is DARPA's High Performance Knowledge Bases (HPKB) project (Cohen et al. 1998). They tested knowledge bases with different sets of axioms to answer the same set of queries. Thus the "completeness" of the system depends as much on the axioms the system uses as it does on the system's inferential capability. By contrast, we evaluate systems on identical axioms on data.

There has been some effort to benchmark description logic systems (Elhaik, Rousset and Ycart 1998, Horrocks and Patel-Schneider 1998). Their benchmark data consist of TBoxes and/or ABoxes, which can be seen as the Semantic Web counterparts of ontologies and instance data respectively. In the work of Elhaik, Rousset and Ycart, the ABox is randomly generated. However, unlike our benchmark data, the ABox is not customizable and repeatable. They also generate the TBox randomly while our benchmark is based on a realistic ontology. In Horrocks and Patel-Schneider's work, they use both artificial and realistic TBoxes and use synthetic ABoxes. Since like the ATP competitions, their emphasis is on evaluating the reasoning algorithms, they assume sound and complete reasoners. As a result, they have not been able to test the systems with increased sizes of ABoxes due to their poor performance.

Conclusion

We described our approach to benchmarking Semantic Web KBSs. We have shown that our work contributes to address new AI problems that are represented by the Semantic Web in the setting of benchmarking. Moreover, we have discussed how our work could be applied to general AI research in knowledge representation. The underlying ideas and methodologies could serve as the key to answer the general questions of how to evaluate the scalability of the system; how to evaluate the potential performance tradeoff in the system; and how to compare systems that are very different in their development philosophy and design choice. Furthermore, since many AI systems depend on a knowledge base component (e.g. intelligent agents, natural language processing systems, etc.), this work can be used to evaluate these components and aid in improving the scalability of these systems.

Acknowledgements: This material is based upon work supported by the National Science Foundation (NSF) under Grant No. IIS-0346963.

References

Borgida, A. 1992. *From Type Systems to Knowledge Representation: Natural Semantics Specifications for Description Logics*. Intl. J. of Intelligent and Cooperative Information Systems 1(1): 93–126.

Broekstra, J. and Kampman, A. 2002. *Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema.* 1st Intl. Semantic Web Conference (ISWC2002).

Cohen, P. et al. 1998. *The DARPA high-performance knowledge bases project*. AI Magazine 19(4):25–49.

Ding, L. et al. 2005. *Search on the Semantic Web*. IEEE Computer 10(38):62–69.

Elhaik, Q., Rousset, M.C. and Ycart, B. 1998. *Generating Random Benchmarks for Description Logics.* 1998 Description Logic Workshop (DL'98).

Guo, Y., Pan, Z. and Heflin, J. 2004. *An Evaluation of Knowledge Base Systems for Large OWL Applications.* 3rd Intl. Semantic Web Conference (ISWC2004).

Horrocks, I. 1998. *The FaCT System*. Automated Reasoning with Analytic Tableaux and Related Methods Intl. Conference (Tableaux'98).

Horrocks, I. and Patel-Schneider, P. 1998. *DL Systems Comparison*. 1998 Description Logic Workshop (DL'98).

Kopena, J.B. and Regli, W.C. 2003. *DAMLJessKB: A Tool for Reasoning with the Semantic Web*. 2nd Intl. Semantic Web Conference (ISWC2003).

Lesser, V. et al. 2000. *Big: An agent for resource-bounded information gathering and decision making*. Artificial Intelligence, 118(1-2):197–244.

Pan, Z. and Heflin, J. 2003. DLDB: *Extending Relational Databases to Support Semantic Web Queries*. 1st Intl. Workshop on Practical and Scalable Semantic Systems.

Patel-Schneider 1989. *A Four-value Semantics for Terminological Logics*. Artificial Intelligence 38(8): 319–351.

Sutcliffe G and Suttner C. 2004. *The CADE ATP System Competition*. Automated Reasoning: 2nd Intl. Joint Conference, IJCAR 2004.

Tsarkov, D. and Horrocks, I. 2003. *DL reasoner vs. first-order prover.* 2003 Description Logic Workshop (DL2003).

van Harmelen, F. 2002. *How the Semantic Web will change KR: challenges and opportunities for a new research agenda*. The Knowledge Engineering Review 17(1): 93–96.

Wang, S., Guo, Y., Qasem, A. and Heflin, J. 2005. *Rapid Benchmarking for Semantic Web Knowledge Base Systems.* 4th Intl. Semantic Web Conference (ISWC2005).

Zilberstein, S., and Russell, S. 1995. *Approximate Reasoning Using Anytime Algorithms.* In Imprecise and Approximate Computation, Kluwer Academic.