An Ontological Approach for Querying Distributed Heterogeneous Information Systems Under Critical Operational Environments

Atif Khan, John Doucette, and Robin Cohen

David R. Cheriton School of Computer Science, University of Waterloo
{a78khan, j3doucet, rcohen}@uwaterloo.ca

Abstract. In this paper, we propose an information exchange framework suited for operating in time-critical environments where diverse heterogeneous sources may be involved. We utilize a request-response type of communication in order to eliminate the need for local collection of distributed information. We then introduce an ontological knowledge representation with a semantic reasoner that operates on an aggregated view of the knowledge sources to generate a response and a semantic proof. A prototype of the system is demonstrated in two example emergency health care scenarios, where there is low tolerance for inaccuracy and the use of diverse sources is required, not all of which may be known to the individual generating the request. The system is contrasted with conventional machine learning approaches and existing work in semantic data mining (used to infer knowledge in order to provide responses), and is shown to offer theoretical and practical advantages over these conventional techniques.

1 Introduction

As electronic information systems become mainstream, society’s dependence upon them for knowledge acquisition and distribution has increased. Over the years, the sophistication of these systems has evolved, making them capable of not only storing large amounts of information in diverse formats, but also of reasoning about complex decisions. The increase in technological capabilities has revolutionized the syntactic interoperability of modern information systems, allowing for a heterogeneous mix of systems to exchange a wide spectrum of data in many different formats. The successful exchange of raw information is, however, only the first step towards solving the bigger semantic challenge of information exchange. This is analogous to the “ontology challenge” defined by [15].

In recent years a focused effort in the semantic web domain has resulted in technological advancements, providing sophisticated tools for intelligent knowledge representation, information processing and reasoning. Domain specific knowledge can be managed by utilizing a diverse set of ontological solutions, which
capture key domain concepts and the relationships between them. Knowledge regarding the domain can then be shared by publishing information in a domain specific ontology. A semantic reasoning engine can then be applied to a knowledge-base to answer complex user queries. The semantic reasoning process allows for enhanced knowledge discovery that may not be possible during analysis of raw data. Latent relationships can be discovered by applying inference rules to the ontological knowledge-base.

Although the premise of the semantic web technology is sound in principle and the use of an ontology can significantly enhance how users consume and process information, practical implementations all but demands that the distributed heterogeneous knowledge be represented by local ontological representations [13]. Consequently, it is still difficult to share knowledge across diverse heterogeneous sources to answer specific questions. Furthermore, under adverse conditions (i.e. constraints on time, communication and/or knowledge), the usefulness of the aggregated data decreases sharply, since human agents are required to (manually) process and reason with the data.

For example, in a healthcare setting, a physician may need to consult various medical information systems in order to determine the best possible solution for a patient. Given ideal conditions, a physician will be able acquire and process information from various systems and make the ideal diagnosis. However, if the same scenario is now constrained by the available time, communication bandwidth and the skill level of the physician, the same quality of medical care may not be possible.

Motivated by this, we propose a framework where a user will pose questions to the system, rather than aggregate knowledge locally. The framework will (a) process the user query, (b) aggregate information from various sources, (c) create a semantic representation of the aggregated data, and (d) process information using a semantic reasoner. Each answer generated (in response to the user query) is backed up by a semantic proof. The semantic proof has the desirable property that it can be validated by any third party. Our approach does not require exchange of large data-sets to make a decision, and consequently is more suitable for the above mentioned adverse scenarios.

2 Proposed Solution

We propose a framework for reliable information exchange between distributed heterogeneous parties, using semantic web technologies under adverse conditions. We observe that under normal circumstances, such an exchange can easily be accomplished using existing techniques. However, these techniques fail to be of practical use under adverse situations. For example, consider the following time and information constrained setting: A patient is in a critical life threatening situation, and is being treated by an emergency response (EMR) team member. Under these conditions the EMR team member may not be able to provide the best personalized care, because of difficulty accessing patient medical records
in a timely manner, or lack of the skills required to correctly interpret those records.

Our proposed framework builds on top the semantic web technologies. We use ontological models for knowledge representation. We acknowledge the fact that diverse heterogeneous information will be represented by an array of local or domain ontologies. Therefore, our framework provides support for working with multiple data-sets represented by different ontological models. Given the almost infinite amount of information in the world, we utilize a problem context to identify and limit the amount of knowledge that needs to be processed. We create a closed world problem specific information model using this context. We utilize a semantic reasoner that takes as its input the knowledge-base, inference rules and the user query. The reasoner generates a two part result-set, where the first element is the answer to the user query and the second element is a semantic proof.

We will now discuss the details of the various components of our proposed framework along with some examples.

2.1 System Architecture

We present a flexible architectural style for our proposed framework. Previous approaches utilizing similar frameworks tend to be domain specific (e.g. [17]). In contrast, our approach is domain independent. Let us examine the salient components of our design. Please refer to Fig 1.

System Interface The system interface component facilitates interaction by allowing an agent to pose a query to the system. The agent may also provide a query specific context. We provide support for two types of agent communications based on the following two agent classifications (i) a computational agent (CA) – represents an artificially intelligent automated system and (ii) a human agent (HA) – representing a human being. The first type of agent communication is between two CAs. A local CA receives a query (and a context) from a remote CA. This type of communication represents distributed automated system interacting with each other. The second type of communication utilizes a local CA and a remote HA. This allows human beings to pose queries to a local system. For each query, the interface receives a response from the reasoning module, and forwards this response to the remote agent.

The system interface component provides a query-able abstraction around the heterogeneous knowledge stores, the actual data (that is utilized for answering the query) does not have to be transmitted. This characteristic of the framework facilitates knowledge sharing under adverse conditions.

Knowledge-Representation The knowledge-representation component of our framework follows a multi-tiered design that is capable of accepting data from
a wide array of heterogeneous sources. It also utilizes the problem-context (generated from the user query context) to limit the amount of data which must be processed to answer the query.

The raw data layer provides a useful abstraction to deal with all non-semantic data sources. These data-sources are composed of structured data (such as in the case of distributed relational database systems) and semi-structured data (such as content repositories – web pages). We assume that this raw data does not have any semantic capabilities built into it.

Information from the raw data layer is then annotated using appropriate ontologies. This semantic data layer provides the appropriate abstraction. It is important to note that we do not constrain the choice of the ontologies used. The main goal here is to be able to convert raw data into its semantically equivalent representation. The semantic data layer is also capable of incorporating data from other semantic data repositories.

The problem-specific semantic layer provides a normalization of the semantic data layer. The main goal of this layer is to provide mappings between various ontological representations of the data in use. For example a single semantic concept (such as name) that may be defined by different ontologies can be normalized and represented by a concept from a single consistent ontology.
**Reasoning and Inference** The reasoning layer is responsible for processing the various inputs from other modules such as the semantic query (representing the initial user query), the inference rules, and the knowledge-base from problem-specific semantic layer. It utilizes a semantic reasoner [27] to reason about the user query over the selected knowledge-base. The reasoner generates a two part result-set. The first element of the result-set contains the answer to the user query. The second element contains a semantic proof in support of the response.

A semantic proof has the desirable property that it can be validated by any party. In a heterogeneous multi-agent distributed environment, knowledge will be in temporal flux. Therefore, the same query may not result in the same answer at a different time. Hence, having a semantic proof generated for each user query allows us to validate an answer against the knowledge-base representation (that was aggregated by the problem-specific semantic layer) at any given instance in time by any party.

**Motivation** In this section we consider two simple scenarios for knowledge sharing under adverse conditions, constrained by lack of time and lack of knowledge. The purpose of these examples is to highlight the various components of our proposed architecture and their interactions with one another. Fig 3 describes a semantic model capturing the high level entities for a medical scenario. This semantic model represents the normalized view of the information gathered from various distributed sources. Furthermore, the model describes not only the entities, but also the semantic relationships between these entities.

The main entities defined in our model are patients, health care providers, drugs, diseases and various medical conditions. For the sake of simplicity, we define various simple relationships between these entities. The main relationship is the IS_A relationship (sometimes called “subsumption”). For example a doctor IS_A health care provider which IS_A person. Similarly Insulin IS_A allopathic drug which IS_A drug. In addition to the IS_A relationship, we also define several other varieties of attribute-value relationship. For example the disease Ulcer has a condition called Bleeding, the drug Nitroglycerin has contra indication to
the drug Viagra. Using the triple notation [22] we capture the semantic model in a triple-store.

**Scenario 1** Consider an emergency response team member who would like to administer Warfrain (an anticoagulant drug) to Alice in order to treat her for potential blood clotting. Alice is currently early in her pregnancy. The EMR member has had no past interactions with Alice, and is not aware of her medical condition and history. We add the following two constraints to this scenario to incorporate the time and knowledge (adverse) factors.

- The current conditions prevent the EMR person from accessing and reviewing Alice’s medical records.
- Alice’s blood clot condition needs to be treated urgently.

Instead of aggregating information related to this scenario (such as Alice’s medical records, drug interaction guidelines and such), the EMR person would launch a natural language query such as “can Alice be given Warfrain?” against a medical information system based on our framework. The system would identify Alice and Warfrain, and would compile the required information from various heterogeneous sources. The compiled knowledge is then translated into its semantic representation. Fig. 4 shows a simplified contextual model based on the global knowledge store presented in fig. 3. The semantic reasoner will consume this information along with the rules and semantic (user) query, and will generate a result and a proof as follows:
User Query

{:Alice :canNotBeGiven :Warfrain}.

Inference Rule

\{
\text{?PATIENT :condition ?CONDITION.} \\
\text{?DRUG :contraIndication ?CONDITION.} \}
\rightarrow \{\text{?PATIENT :canNotBeGiven ?DRUG}\}.

Semantic Reasoning & Proof

\{ 
\{ {:Alice :condition :Pregnancy} e:evidence <knowledge-base#_27> \} \}
\cup \{ 
\{ {:Warfrain :contraIndication :Pregnancy} e:evidence <knowledge-base#_22> \} \}
\rightarrow \{ 
\{ {:Alice :canNotBeGiven :Warfrain} e:evidence <rules#_9> \} \}
\text{Proof found in 3 steps (2970 steps/sec) using 1 engine (18 triples) }.

Based on the facts and the inference rules, the semantic reasoner concludes that Alice can not be given Warfrain since she is pregnant and the Warfrain is a contra indication to Pregnancy.

Fig. 4. Scenario 1: Should Alice be given Warfrain?

Scenario 2 Alice, who is pregnant, is suffering from a urinary tract infection. Trimethoprim-sulfamethoxazole is an antibiotic, commonly prescribed to treat
urinary tract infections. Assuming the same constraints as scenario 1, the EMR person (who is treating Alice) queries the system as follows:

User Query

```
\text{Alice} :canNotBeGiven :TrimethoprimSulfamethoxazole.
```

Inference Rule

```
\{ \text{?PATIENT} :hasDisease \text{?DISEASE}. \\
\text{?DISEASE} :treatedBy \text{?DRUG}. \\
\text{?DRUG} :contraIndication \text{?CONDITION}. \\
\text{?PATIENT} :condition \text{?CONDITION.} \} \Rightarrow \{ \text{?PATIENT} :canNotBeGiven \text{?DRUG}. \}
```

Semantic Reasoning & Proof

```
\{ \{ \text{:Alice} :hasDisease \text{:UrinaryTractInfection} \ e: evidence < knowledge-base#_36 > . \\
\text{:UrinaryTractInfection} :treatedBy \text{:TrimethoprimSulfamethoxazole} \ e: evidence \\
< knowledge-base#_31 > . \\
\text{:TrimethoprimSulfamethoxazole} :contraIndication \text{:Pregnancy} \ e: evidence \\
< knowledge-base#_26 > . \\
\{ \text{:Alice} :condition \text{:Pregnancy} \ e: evidence < knowledge-base#_35 > \} \Rightarrow \\
\{ \{ \text{:Alice} :canNotBeGiven \text{:TrimethoprimSulfamethoxazole} \ e: evidence \\
< rules#_12 > \}. \\
\}
```

Fig 5 represents the simplified semantic model that is used for illustrating scenario 2.

The two scenarios discussed above have been kept simple for ease of understanding. A more realistic knowledge-base would be quite rich in semantic concepts and a large number of relationships between the concepts. Similarly there will an array of rules defined to provide the required level of inferencing capabilities for a complex semantic model.

3 Related Work

In this section we establish both theoretical and practical concerns motivating the use of ontologies in knowledge querying, and discuss previous work incorporating ontologies into knowledge querying.
3.1 Ontology-Free Approaches to Querying

There is considerable recent work suggesting that conventional querying techniques, though extremely powerful, might not be suitable for use in environments where queries are frequent, time-dependent, and arbitrarily complex. The principal reason for this is that, in the absence of semantic reasoning and inference rules, all information available for querying must be available in explicit form. This poses a problem in domains where there exists an enormous amount of information, precluding the possibility of explicit codification. For example, in the medical domain, there are literally millions of codified relationships between various concepts, but the hierarchical nature of these relationships means that the number of implicit relationships can be much higher. For example, bacterial pneumonia has all the attributes and relationships of a bacterial infection and a lung infection (explicit), but is also, implicitly a type of infection, and a type of lung disease.

This motivates the use of machine learning techniques as a possible method of answering ad-hoc queries to a database which may not encode all possible relationships. By taking a sufficiently large sample of the data, it may be possible to infer the answer to a user's query. For example, if a user asks whether a particular patient can be given a drug, a predictive classification system could be dynamically constructed and utilized to answer the query.

3.2 Motivations for Ontological Approaches to Querying

There are both theoretical and practical motivations for avoiding the use of machine learning techniques to answer questions in the way described above. First, let us consider the theoretical issues. Many machine learning algorithms, including popular decision trees (e.g. C4.5, ID3 [23]), maximum margin classifiers (e.g.
Support Vector Machines [7]), and clustering techniques (e.g. K-Nearest Neighbors [9]), operate by phrasing queries as optimization problems. For example, if a doctor wants to know whether their patient is likely to experience an adverse reaction to a drug, then a system might collect a large sample of patient records and use them to build a classification model by minimizing the error rate in predicting the outcome of drug administration to those patients. Although machine learning algorithms are often very effective in practice, there are theoretical reasons to suppose they might be less useful in time-critical domains where many arbitrarily formed queries are being made. "No Free Lunch" theorem (NFL) [29] shows that all optimization techniques are expected to produce identical mean performance across a set of arbitrary queries, in the absence of knowledge specific to each query being incorporated into the algorithms. This somewhat counter-intuitive result suggests that, over a large set of possible queries, as might occur in an emergency scenario, no conventional machine learning technique is likely to answer all queries better than using completely random optimization strategies. In a critical scenario like the ones used as an example throughout this paper, the possibility of receiving a poor result might be too large a risk (whether perceived or real) for users to trust the system’s answers.

Second, there are practical considerations. The principal practical consideration in our scenario of interest is the opacity of the answers obtained using conventional query techniques. Continuing with our example above, what the doctor receives in response to a query about adverse reactions to a drug is a classification model based on a sample of patient data. The understandability of these models to computational laypeople varies from model to model. A support vector machine for example, is practically impossible for a layperson to understand, since it operates by building the maximally separating hyper-plane for a high-dimensional extrapolation of the given data. When the doctor asks "Why does the system believe my patient will have an interaction?", she may not trust a system which answers "I put your patient’s record into a 500 dimensional space, and it fell on this side of a line". This is true even if the system is highly reliable, because human users may have concerns about the ethics of entrusting life threatening decisions to a "black box". The system cannot easily explain its decision in terms of medical conditions and the relationships between them, and so it is impossible to tell whether the answer provided is based on sound reasoning, or an unfortunate hiccup in the algorithm’s usual consistency. Furthermore, there may be a valid reason to distrust systems built from patient data, namely the often quoted “Garbage In Garbage Out” (GIGO) principle. If the relationships present in the data-set do not include the specific situations of this patient, some machine learning methods may not be able to detect the dissimilarities, and may falsely report high confidence in their results.

Although these practical concerns can be somewhat mitigated (e.g. the use of rule-based models can reduce opacity; the use of local membership functions can mitigate GIGO concerns), there are comparatively few ways to address the theoretical problems posed by No Free Lunch. While certain techniques, like competitive co-evolutionary methods [30], can be used to provide “free lunches”
(i.e. algorithms with expected performances better than random approaches), these techniques have not seen widespread adoption.

3.3 Previous Ontological Approaches to Querying

There has been considerable previous work utilizing ontologies for answering queries, but the general focus is on preprocessing of queries to facilitate conventional machine learning techniques. This is a reasonable approach insofar as it obviates the NFL issues described above by introducing domain specific knowledge into the optimization process. In medicine, for example, there has been a focus on isolating the queries used by doctors the most frequently, and preprocessing them using semantic information. Doctors questions tend to focus on finding the underlying cause of a set of symptoms or finding the appropriate medication and dose for a particular patient [8,12]. By utilizing ontological information, previous researchers have created systems capable of automated Contextualization of doctor queries. For example, a doctor whose patient has type I diabetes would have queries regarding that patient and “diabetes” automatically translated to instead include “Type I Diabetes” [21]. An alternative approach considers the incorporation of meta-data into search queries, which can be utilized to return more relevant documents during information retrieval [11]. Finally, recent research in question answering systems utilizes ontologies to translate doctors’ questions into lists of relevant terms for an ordinary search engine [14].

Outside of the medical domain, ontologies have been used to facilitate the design of query answering processes [3]; to preprocess highly abbreviated data into a more understandable format [10]; and to post-process the models produced by conventional machine learning processes [4]. There is, however, little work examining the possibility of replacing conventional machine learning algorithms entirely, and utilizing a semantic reasoner in their place.

The use of a semantic reasoner in place of a conventional machine learning algorithm to answer search queries offers several immediate advantages. First, because a semantic reasoner does not rely on optimization to construct a predictive model, it is not subject to the problems posed by the No Free Lunch theorems for optimization. This eliminates the need for extensive incorporation of a priori knowledge by the end user, as in [11]. Second, the opacity problem is solved by the ability of the system to both provide a proof of its answer (i.e. the chain of reasoning used to determine the answer), and to formulate that proof in terms easily understood by a layperson (i.e. via conversion of triple formatted data into simple natural language statements). For example, if our doctor wishes to ask “Why does the system believe my patient will have a reaction to this drug?”, instead of being told, somewhat tautologically, that their patient fits the system’s model of patients who had reactions, or that their patient is similar to other patients who had reactions, the doctor can be provided with a patient-specific proof based on sound medical understanding. If the doctor wants to know whether she can administer vascular constrictors, the system could report:
1. The patient has high blood pressure.
2. Patients with high blood pressure should not usually be given vascular constrictors.
3. By 1 and 2, the patient should probably not be given vascular constrictors.

It is this advantage that is most critical in scenarios where time is short and individual knowledge may be low. By providing reasoning, the system allows users to check that the assumptions and data present in the system are valid. As an additional example, consider the survivor of a natural disaster who wants to know whether it is safe to cross a bridge that may have been damaged, but fails to input critical information (e.g. “The bridge appears to have hairline fractures.”) along with their query. While a conventional model may simply report that the bridge is safe, or show a potentially very complex rule used to make that determination, a semantic reasoner might provide the proof:

1. The bridge does not have any other heavy objects on it.
2. The bridge does not appear to be bent.
3. The bridge does not appear to be cracked.
4. Bridges which do not appear to be bent or cracked, and are free of heavy objects are safe to drive on.
5. By 1, 2, 3, and 4, the bridge is safe.

Notice that while it is possible to produce similar summaries using conventional techniques via rule extraction, rules may be based on correlations from the data-set that do not have clear causal relationships. For example, it might be the case that being overweight is strongly correlated with reactions to vascular constrictors in a particular data-set, and coincidentally, it happens that all persons with high blood pressure are also overweight, but only a small number of overweight persons do not have high blood pressure. In this case, a conventional data mining technique might report the use of a rule like:

1. The patient is overweight.
2. Overweight people have more frequent reactions to the vascular constrictors.
3. The patient should not be given vascular constrictors.

Although this is based on a true correlation in the data, it is not based on sound medical understanding. Likewise, in our bridge example, it might be true that the data-set contains patterns which specify that the presence of water under a bridge is the single most important factor in predicting whether it will be safe in the aftermath of a natural disaster. Again, this relationship is not causal. The presence of water under a bridge might be an indication that the bridge is of a type that is less structurally sound, but the presence of water itself does not play a factor in whether a particular bridge is safe. Consequently, although such a rule might perform very well in practice, users might be hesitant to accept the opinion of a system which produces correlations that do not appear to have a causal basis.
4 Future Work

Future work will take two directions. First, we plan to implement and benchmark a prototype of our system, and compare its performance with that of a system based around conventional machine learning techniques for question answering. Second, we plan to extend the framework by overlaying probabilistic models onto the ontological model, so that the system can provide a more precise answer to a users’ queries. For example, a user who reports cracks in a bridge might be told that there is a 60% chance of bridge failure, rather than simply being told that the bridge will collapse if they drive over it. The drawback of this system is the “curse of dimensionality” which arises when there are many possible combinations of factors that have different interactions. For example, maybe a bent bridge has a 30% chance of collapsing, but a bent and cracked bridge has a 99% chance of collapsing. The problem grows exponentially worse as additional factors are added, and each combination of factors in turn must be considered.

To avoid this problem, we plan to consider the introduction of heuristic techniques for providing estimated probabilities. For example, we might have the system take a random sampling of past bridges with both characteristics, and produce an observed probability estimate. Alternatively, the system could provide “reasonable” bounds in the absence of additional information by assuming no interaction and a positive interaction of strength proportionate to the criticality of the task. Thus, if the bridge is only 3ft off the ground, estimates of the risk would tend to be liberal (i.e. smaller interaction estimates) than if the bridge is 300ft off the ground. Neither scheme is ideal, and experimental validation might be required to determine appropriate estimates of risk.

5 Conclusion

In this paper we present a proposal for a general purpose ontology-based information exchange framework, intended for use in time critical, knowledge sparse scenarios. The system utilizes ontologies to retrieve contextually relevant facts from external data sources; reason about those facts in the context of a problem-dependent rule base; and produce both answers and human readable proofs relevant to user queries.

The system is demonstrated through two example scenarios with a prototype, and contrasted with existing work on semantic data mining (which tends to focus on pre- and post-processing, rather than rule discovery and query answering), and conventional, non-semantic machine learning approaches. Our system eliminates the problems posed by the No Free Lunch theorem for optimization [29], and provides transparent answers which are easily understood by computational laypersons. Future work will focus on the implementation of a fully functional system, user studies of the system’s effectiveness as compared with conventional techniques, and incorporating probabilistic reasoning into the model.


22. Notation 3 (n3): A readable RDF syntax. Website.


27. Joe De Roo. Euler proof mechanism. Website.


