ONTOLOGY-DRIVEN QUESTION ANSWERING AND ONTOLOGY QUALITY EVALUATION

by

SAMIR TARTIR

(Under the Direction of Ismailcem Budak Arpinar)

ABSTRACT

As more data is being semantically annotated, it is getting more common that researchers in multiple disciplines to rely on semantic repositories that contain large amounts of data in the form of ontologies as a compact source of information. One of the main issues currently facing these researchers is the lack of easy-to-use interfaces for data retrieval, due to the need to use special query languages or applications. In addition, the knowledge in these repositories might not be comprehensive or up-to-date due to several reasons, such as the discovery of new knowledge in the field after the repositories was created. In this dissertation, we present our SemanticQA system that allows users to query semantic data repositories using natural language questions. If a user question cannot be answered solely from the ontology, SemanticQA detects the failing parts and attempts to answer these parts from web documents and plugs in the answers to answer the whole questions, which might involve a repetition of the same process if other parts fail.

At the same time, with the large number of ontologies being added constantly, it is difficult for users to find ontologies that are suitable to their work. Therefore, tools for evaluating and ranking the ontologies are needed. For this purpose, we present OntoQA, a tool that evaluates ontologies related to a certain set of terms and then ranks them according a set of metrics that captures different aspects of ontologies. Since there are no global criteria defining how a good ontology should be, OntoQA allows users to tune the ranking towards certain features of ontologies to suit the need of their applications. OntoQA is not only useful for users trying to find suitable ontologies, but for ontology developers who are looking for measures to evaluate their product.

INDEX WORDS: Question Answering, Quality Evaluation Knowledge Discovery, Entity Spotting, Semantic Web, Ontology, Web Search, OWL, RDF

ONTOLOGY-DRIVEN QUESTION ANSWERING AND ONTOLOGY QUALITY EVALUATION

by

SAMIR TARTIR

B.Sc., University of Jordan, Jordan, 1998

M.S., University of Jordan, Jordan, 2002

A Dissertation Submitted to the Graduate Faculty of the University of Georgia in Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2009

ONTOLOGY-DRIVEN QUESTION ANSWERING AND ONTOLOGY QUALITY EVALUATION

by

SAMIR TARTIR

Major Professor: Ismailcem Budak Arpinar

Committee: John A. Miller
Liming Cai

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2009

DEDICATION

I would like to dedicate this thesis to my parents, Yacoub and Entesar Tartir, and to my wife Nadin, and daughter Leen without whose support this work would not have been possible.

ACKNOWLEDGEMENTS

First of all, I would like to thank God for giving the power to continue with this work, without His help, none of this could have been possible.

I would like to also thank my advisor I. Budak Arpinar for his advice, guidance and support. In particular, I thank him for giving me the opportunity for mentoring several students pursuing their Master's degree. I would also like to thank members of my doctoral committee, Professors John A. Miller, and Liming Cai. I would like to especially thank Professor Miller for his continued support, guidance and invaluable insights that led me to think of him as unofficial co-advisor.

I have been lucky to be part of a group consisting of many active students. The opportunity to work with a variety of enthusiastic young researchers has been invaluable. I appreciate the opportunities for research/collaboration as well as discussions/interactions with a variety of people, mainly Boanerges Aleman-Meza, Matthew Eavenson, Maciej Janik, and other members/alumni of the LSDIS Lab.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Large amounts of data in many disciplines are continuously being added to semantic repositories as a result of continuing research in different scientific fields, and it is becoming an increasing challenge for researchers to use these repositories efficiently and at the same time cope with this fast pace of the introduction of new knowledge [30]. For example, the National Library of Medicine's MeSH (Medical Subject Heading) vocabulary is used for annotation of scientific literature. Efforts in industry [51] as well as those by scientific communities (e.g., Open Biological Ontologies[1]), which lists well over eighty ontologies) have demonstrated capabilities for building large populated ontologies. Additionally, metadata extraction and annotation in web pages has been addressed earlier and proven scalable [8][17][25]. Although publishers of such ontologies try to keep up with the pace of that knowledge expansion, it will be difficult for these semantic repositories to always contain the up-to-date knowledge that exists, for example, in published journal articles or online repositories before these ontologies get updated with the new knowledge.

An ontology is an explicit specification of a conceptualization [21] that represents a set of concepts within a domain and the relationships between those concepts and is encoded using one of the ontology languages, such as OWL [3] or RDF[2]. Ontologies usually encode concepts and relationships of a domain defined in a schema, and in many cases specific domain instances or objects. Figure 1 below shows a sample ontology [11] that includes six concepts, two relationships and 6 instances of various types.

[1] http://obo.sourceforge.net
[2] http://www.w3.org/RDF/

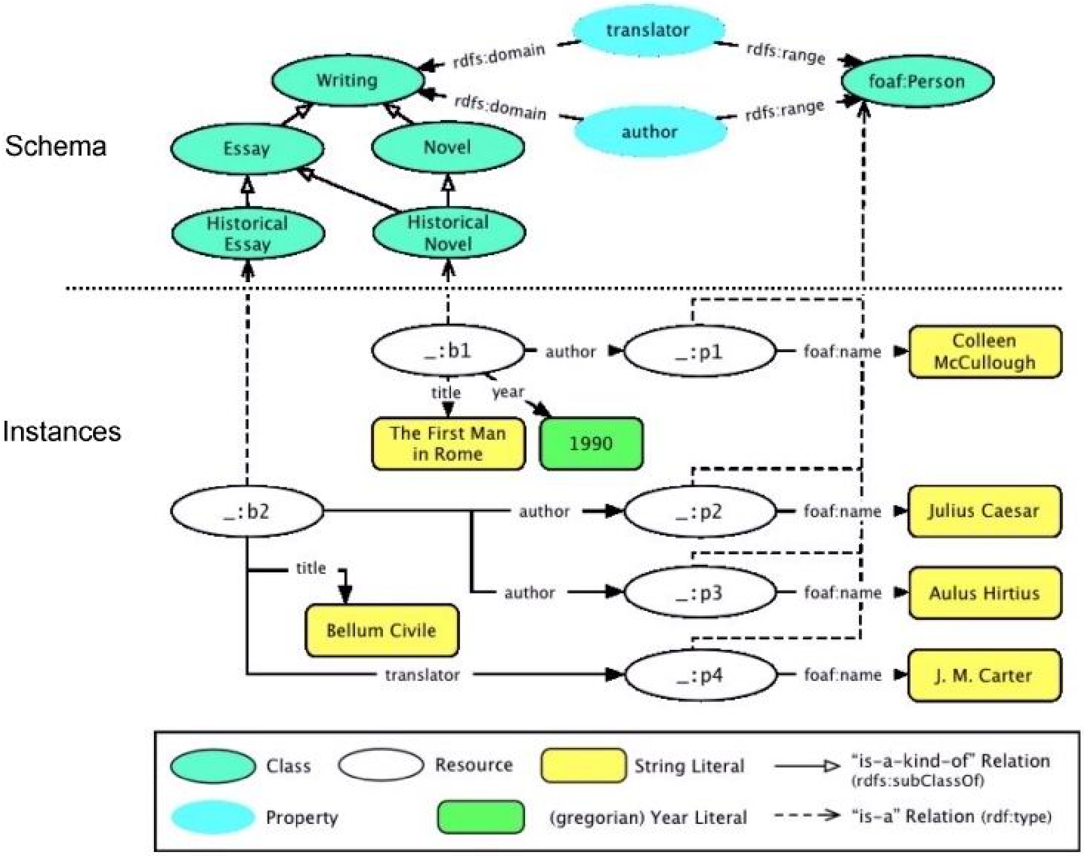


Figure 1: Sample Data in as a Simplified RDF Graph

With the introduction of such a great volume of semantic data in the form of ontologies, the need for tools and methods to evaluate and describe contents of these ontologies is growing. For example, in a scenario where an ontology knowledgebase building process is performed automatically by extracting instances from web pages automatically, it is usually desired to have a measure of how well the extraction process performed in covering the domain knowledge, or whether the schema that is built for a system was built to deep, or with too few relationships. Being able to get a glimpse of the contents of ontologies is also needed by users of ontologies, who can be in situations where they have multiple ontologies that can satisfy their needs and they need to choose the most appropriate, without having to look into the contents of each ontology. Our research on ontology evaluation [59][60][61] has covered different aspects of the design of the ontology (e.g., depth), and the population of the ontology (e.g., classes that are populated than others, relationships that are mostly used) to give users detailed information about the ontology.

After guaranteeing that the quality of an ontology is satisfactory, it can be used in applications including answering user questions, a process that usually requires a good coverage of the domain both in concepts and relationships between them and in actual instances that represent real-world facts.

Question Answering is defined as an interactive human computer process that encompasses understanding a user information need, typically expressed in a natural language query; retrieving relevant documents, data, or knowledge from selected sources; extracting, qualifying and prioritizing available answers from these sources; and presenting and explaining responses in an effective manner [39]. Providing answers to user questions is always improving. In May 2009, Wolfram|Alpha[3] was introduced as a "computational knowledge engine", rather than a simple search engine as its interface indicate, and it is claiming to answer some types of user questions, in a process that goes deeper than simple document retrieval.

Ontologies can play an important role in question answering. In addition to containing answers to user questions, ontologies help in unifying terms in a domain, as they are usually built with as much domain expert consensus as possible, thus using terms included in an ontology in a question will make answering it easier. In addition, utilization of an ontology will allow a question answering system better facilities to recognize entities and disambiguate them as an ontology represents a single domain usually, where an entity has a single meaning, or use the surrounding entities to disambiguate between the different meaning when there is more than one meaning. Finally, ontologies can provide an insight on the answer of the question. For example, if the question is about an advisor of a graduate student in a university domain, a relationship that is defined in the ontology between graduate students shows that the other type of the relationship should be of type "professor", and this will help filter out unrelated answers, especially when extracting answers from web documents.

[3] http://www.wolframalpha.com

Extracting answers for a user – who can come from multiple disciplines – writing questions in English from such ontologies usually requires understanding the contents of the ontology, which entails understanding an ontology language, or query these ontologies by building complex queries using one of the ontology query languages, such as SPARQL[4], instead of being able to present their questions as they think about them, in natural language. Such prerequisites are some of the major reasons driving such users from fully utilizing the large amounts of knowledge in ontologies, forming major problems for the whole Semantic Web Initiative [7][17].

The method proposed in this dissertation is focused on the process of answering natural language questions by utilizing domain knowledge stored in such ontologies to extract answers to these questions from the ontology, as well as one or more web documents when the question can not be answered from the ontology alone.

Natural language question answering has been addressed before. A number of techniques attempt to process and answer questions without the background knowledge that an ontology might contain, making such techniques produce imprecise answers. Other techniques rely on an ontology totally, which limits their answering capabilities to only what the ontology contains. Others techniques allow the user to enter the query using a predefined set of templates which results in an increased complexity for regular users. Others allows the user to enter his/her question as a bag of keywords that will be treated equally with no distinguishing between what is a type from what is a relationship, also resulting in imprecise answers. Also, most question answering techniques that uses web documents as a source of answers usually return a document set as an answer instead of returning an answer or a list of candidate answers to the user. Expert systems in artificial intelligence have also address the issue of question answering, but most approaches require a series of questions and answers when attempting to solve a problem, and are less adaptive to changing environment unless the knowledgebase is constantly updated.

[4] http://www.w3.org/TR/rdf-sparql-query/

Our approach [58] first utilizes ontological knowledge by assisting users in building their questions as they type, by presenting them with relevant suggestions extracted from the ontology based on previous input. For example, if a user was using the ontology in Figure 1, and he enters the concept “writing” the system will provide him next with relationships that are defined on that concept, in our case “translator” and “author”. This approach makes it easier for users to use phrases that are domain standards instead of using phrases they make that will understanding the question more difficult. Then, the question is processed using the ontology to extract entities (instances, concepts and relationships) that belong to the domain ontology. This process involves entity spotting, where entered phrases are matched to labels of ontology entities, and to any linguistic alternatives these labels might have.

For example, using the ontoloty in Figure 1 if a user enters the question:

“Who is the writer of Bellum Civile?”

the word “writer” will be matched to the relationship “author” in the ontology using synonym matching. The extracted relationships, concepts, and instances, are then converted to a set of subject-predicate-object triples that will be used to form SPARQL queries, and keyword web searches if needed. In this case, a triple will be generated containing:

<> <wrtier> <Bellum Civile>

This triple is then used to form a SPARQL query that is run against the ontology to find if an answer for whole query can be found in the ontology. In our case, the answer will be returned as the two authors of the essay: Julius Caesar and Aulus Hirtius. If the query fails, that indicates that some parts of the query (some triples) could not be answered only from the ontology by alone. Answers for these triples are extracted from web pages gathered by performing several keyword-based searches. In this case, answers are extracted from these documents and then ranked using a novel measure, the *Semantic Answer Score*, that we devised that extracts the best answer from relevant documents and returns it to the system so it can be used to answer the whole query.

1.1 Contributions

The contributions of this dissertation demonstrate the benefits of combing natural language processing techniques to query ontological knowledge and web documents. The necessary components to make this possible include new techniques as well as the use and adaptation of earlier techniques using natural language parsers. The contributions of this dissertation are as follows:

(1) A flexible semantic question answering approach that can be applied to different domains simply by changing the base ontology without any loss of functionality.

(2) Semantic techniques that allow users the freedom to build natural language questions with the help of a suggestions system that provides suggestions from the ontology based on previous user input by utilizing the relationships, concepts and instances defined in the ontology.

(3) An ontological approach that converts questions asked in natural language to query triples by employing different semantic and linguistic techniques. These triples allow the system to divide the question into smaller parts that can be processed independently from each other.

(4) A multi-source answering technique that combines the ability to extract answers from an ontology, and from one or more web pages if needed. The technique checks whether all triples in the question can be answered from the ontology alone. If some can not, the system extracts answers for each of these triples from web pages relevant to this specific triple, not the whole query. For extracting answers from web documents, the technique employs a novel measure, the *Semantic Answer Score*, which we devised to allow the technique to retrieve the most relevant answers from web documents.

1.2 Context and Scope

The information retrieval research community has addressed the problem of question answering, an area that keeps on evolving, but there are additional challenges and possibilities when Semantic Web techniques are considered. Question answering techniques are developed considering the possibilities offered by the different types of questions that can be asked to the system, the targeted source for answers, and how answers are presented to the user. For example, techniques for natural language question answering from web resources utilize techniques such as synonym expansion and stemming to retrieve as many relevant answers as possible. The methods proposed in this dissertation are intended for answering questions posed in natural language. In addition, the methods require that the *named* entities are mentioned in the questions where such named-entities exist in the ontology being used by the system, otherwise, the question will be outside of the domain of the ontology and can not be answered. The architecture is designed to be able to use arbitrary ontologies. Yet, the methods will perform better when these ontologies are populated ontologies. That is, the ontology should contain a considerable number of named entities interlinked to other entities because the method relies on relationships between entities to extract answers from different resources. Some methods are limited to extracting answers from a single ontology, or from a single web page. Our method answers questions, even if that entails extracting different parts of the answer from different sources, the ontology, or one or more web pages when required. Other approaches exploit the semantics of nouns, verbs, etc. for incorporating semantics in search, for example, by using WordNet [41]. The methods presented in this dissertation exploit semantics of named entities instead.

The challenges in research dealing with question answering include traditional components in information retrieval systems. Due to the nature of many populated ontologies in scientific fields (e.g., biology) that they contain a large number of instances, these need to be properly processed and indexed for efficient operation of the whole system. Other components include fast retrieval of the documents relevant to a query and their ranking. In the work presented in this dissertation, it is also necessary to perform a process of semantic annotation for spotting appearances of named-entities from the ontology in the question and the web documents relevant to any failing triples. The type of challenges involved in techniques that process large ontologies include processing of data that is organized in a graph form as opposed to traditional database tables. The techniques presented in this dissertation make extensive use of graph traversal to determine how entities in an ontology are connected. This is often needed to determine relevant entities according to the paths connecting them. The challenge involved is that ontologies containing over a million entities are no longer the exception [53]. Lastly, other challenges exist in evaluation of the approach. It is typically difficult to devise methods to evaluate many queries in an automated manner. This is due to the difficulty of knowing in advance which parts of the query (triples) have answers in the ontology. In fact, this is a more challenging problem when the search method can differentiate between results that match different named-entities for the same input from user. It would be necessary to know in advance the subset of triples that have can be answered from the ontology. In summary, the challenges involved are in terms of traditional processing of natural language text, as well as processing of large ontologies and its usage for annotation, indexing and retrieval of documents, extracting answers from those documents, and measuring relevance using entities and their relationships for question answering.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter first describes necessary components that are not the main contributions of the dissertation yet are important components of the proposed method for ontology-based multi-source natural language question answering. These components are a populated ontology, semantic annotation of document collection to identify the named entities from the ontology, indexing and retrieval based on keyword input from user. Second, related previous work is described.

2.1 Semantic Web

The Semantic Web is a vision that describes a possible form that the Web will take as it evolves. Such vision relies upon added semantics to content that in the first version of the Web was intended solely for human consumption. This can be viewed from the perspective that a human could easily interpret a variety of web pages and glean understanding thereof. Computers, on the other hand, can only achieve limited understanding unless more explicit data is available. It is expected that the mechanisms to describe data in Semantic Web terms will facilitate applications to exploit data in more ways and lead to automation of tasks. The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.

One of the basic means to explicitly state or add meaning to data is the Resource Description Framework (RDF), which provides a framework to capture the meaning of an entity (or resource) by specifying how it relates to other entities (or classes of resources). Thus, this is a step beyond metadata, in particular, semantic metadata, which can be described as content enriched with semantic annotations using classes and relationships from an ontology [52]. Semantic technologies are gaining wider use in Web applications [54][35][42].

2.2 Ontologies

The development of Semantic Web applications typically involves processing of data represented using or supported by ontologies. An ontology is a specification of a conceptualization [18] yet the value of ontologies is in the agreement they are intended to provide (for humans, and/or machines). In the Semantic Web, an ontology can be viewed as a vocabulary used to describe a world model. A populated ontology is one that contains not only the schema or definition of the classes/concepts and relationship names but also a large number of entities that constitute the instance population of the ontology. That is, not just the schema of the ontology is of particular interest, but also the population (instances, assertions or description base) of the ontology. A highly populated ontology (ontology with instances or assertions) is critical for assessing effectiveness, and scalability of core semantic techniques such as semantic disambiguation, reasoning, and discovery techniques. Ontology population has been identified as a key enabler of practical semantic applications in industry; for example, Semagix (now Fortent[5]) reports that its typical commercially developed ontologies have over one million objects [53]. Another important factor related to the population of the ontology is that it should be possible to capture instances that are highly connected (i.e., the knowledge base should be deep with many explicit relationships among the instances). This will allow for a more detailed analysis of current and future semantic tools and applications, especially those that exploit the way in which instances are related.

In some domains, there are available ontologies that were built with significant human effort. However, it has been demonstrated that large ontologies can be built with tools for extraction and annotation of metadata [27][28][56][61][63]; see [34] for a survey of Web data extraction tools. Industry efforts have demonstrated capabilities for building large populated ontologies [51], which are sometimes called shallow ontologies. *Shallow* ontologies contain large amounts of data and the concepts and

[5] http://www.fortent.com/

relations are unlikely to change, whereas *deep* ontologies contain smaller (or not any) amounts of data but the actual concepts and relations require extensive efforts on their building and maintenance [50].

An ontology intended for question answering calls for focusing on a specific domain where populated ontologies are available or can be built. Ontologies used in our approach need to contain named-entities that relate to other entities in the ontology (i.e., resource-to-resource triples). The named-entities from the ontology are expected to appear in the asked question and in web documents relevant to failed triples. This can be a limitation in certain domains for which ontologies are yet to be created. However, techniques and developments continue for metadata extraction of semantics. For example, a recent work opens possibilities of ontology creation from wiki content [6]. In domains such as life sciences and health-care many comprehensive, open, and large ontologies have been developed. In addition to OBO, UniProtKB[6] and Glyco/Propreo [43] are ontologies with well over one million entities. In domains such as financial services/regulatory compliance [55] and intelligence/defense, a number of non-public ontologies have been developed. Other large ontologies such as TAP [22] and Lehigh Benchmark [23] have also proven useful for developments and evaluations in Semantic Web research. Lehigh Benchmark is a suitable dataset for performance evaluation but it is a synthetic dataset.

2.3 Question Answering

As mentioned earlier, question answering is not a new field. Several approaches have been made with different results. Table 1 below lists some of the main approaches so far.

[6] http://www.pir.uniprot.org

Table 1: Question answering systems

Approach	Input Format	Document / Text Retrieval	Output
EBIMed	Keywords	Local MedLine abstracts	Word-pairs
Cognition	Keywords	Disambiguated text in four domains	Documents
ONBIRES	Category	MedLine abstracts	Sentences
TextPresso	Category	Local scientific literature collection	Sentences
GoPubMed	Boolean query	PubMed articles	Induced ontology
PowerSet	Any	Wikipedia articles	Sentences
PANTO	NL Question	Ontology	SPARQL query
AquaLog	NL Question	Ontology	Answers

As it can be seen, previous approaches have tried different methods to tackle the problem of question answering. Many approaches like EBIMed [48], Cognition[15], ONBIRES [31], and TextPresso [43] for example, any entered input is processed as keywords, without consideration for the semantics of the domain questions need to be answered in. In the case of Cognition, all input text has been manually disambiguated, a huge undertaking that took years to be accomplished. Also, all approaches are answering question from locally stored resources, which can be a limited resource, especially in domains that are constantly evolving. Moreover, most of these approaches are not producing answers; instead they are producing text, which the user has to process to get the answer he is seeking. Additionally, these approaches are very limited to the domains they are built for. In contrast, the approach presented in this dissertation is portable, allowing change of domains simply by switching the background ontology. Also, the developed system accepts input as a question that can be of different degrees of complexity, allowing users to form question to ask specifically what they want. The proposed approach also utilizes the local ontology, and, when needed, information from web documents, instead of always using web documents, or just limiting the answering database to the local knowledge.

Powerset[7] is a commercial question answering system. Although it is showing some promising results, but its lack of understanding of the semantics of the question makes the approach less accurate than others.

The lower half of the table shows approaches PANTO [63] and AquaLog [37] that process NL questions using an ontology. As with the previous *non-semantic* approaches, they are all single-source, either answering a question from the ontology alone, or from a set of web documents without allowing answers from different sources to be integrated together to answer the whole question.

2.4 Ontology Evaluation

In the proposed approach, ontologies form the cornerstone of the whole approach. For the approach to work successfully, the ontology needs to be of a good quality. The work on ontology evaluation presented in Chapter 4 has been successful in this path. Here other approaches on ontology evaluation are presented.

An emerging trend in ontology evaluation is tracking the evolution of ontologies through time. For example, the approach in [47] keeps track of ontology concept evolution through keeping a track of the changes in a version log that can be used to create "virtual versions". The approach also defines a new language Change Definition Language (CDL) that is used in keeping track of the version. The logical approach in [25] goes even further to discover and repair inconsistencies in ontologies across the different versions of the ontology.

A rule-based approach to conflict detection in ontologies is introduced in [5]. In this approach users define what they consider as conflicting rules using RuleML [9] and the approach will then list any cases were these rules are violated. A similar approach has also been used in.[19]

In [38], the authors propose a complex framework consisting of 160 characteristics spread across five dimensions: content of the ontology, language, development methodology, building tools, and usage

[7] http://www.powerset.com

costs. Unfortunately, the use of the OntoMetric tool introduced in the paper is not clearly defined, and the large number of characteristics makes their model difficult to understand.

[45] uses a logic model to detect unsatisfiable concepts and inconsistencies in OWL ontologies. The approach is intended to be used by ontology designers to evaluate their work and to indicate any possible problems.

In [57] the authors propose a model for evaluating ontology schemas. The model contains two sets of features: quantifiable and non-quantifiable. It crawls the web (causing some delay, especially if the user has some ontologies to evaluate), searches for suitable ontologies, and then returns the ontology schemas' features to allow the user to select the most suitable ontology for the application. The application does not consider ontologies' knowledge bases that can provide more insight into the way the ontology is used.

The OntoClean approach in [22] is used for the analysis of taxonomic relationships based on the philosophical notions of rigidity, unity, dependence, and identity.

AKTiveRank authors propose four metrics to rank a group of ontologies related to a set of terms [1]. The metrics are: class match, density, semantic similarity, and betweenness. These four metrics deal with classes that match the search terms in the ontology. The approach then uses a weighted average of the four metrics to produce a rank for each ontology.

Finally, [13] introduces the ODEval tool that can be used to detect possible taxonomical problems in ontologies, such as inconsistency, incompleteness, and redundancy.

Below is a summary of all these systems, and how they are compare to OntoQA.

Table 2: Ontology Evaluation Systems

Approach	User Involvement	Ontologies	Schema / KB
[47]	High	Entered	Schema
[25]	High	Entered	Schema
[5]	High	Entered	Schema + KB
[45]	Low	Entered	Schema
[57]	High	Entered	Schema
[57]	Low	Crawled	Schema
[1]	Low	Crawled	Schema
[13]	Low	Entered	Schema
[22]	Low	Entered	Schema
OntoQA	Low	Both	Schema + KB

Table 2 summarizes some of the main features of the all these approaches. In the user involvement column, it can be seen that the approaches are divided in half in the level of the user involvement required for the approach to successfully achieve its goals. For example, a person using the approach of [47] needs to create a log for each change of the ontology to evaluate any potential problems in the ontology introduced by the change. The second column indicates whether the approach's input ontologies are manually entered by the user or searched for by crawling the web. The last column indicates whether the approach evaluates the ontology schema only or both the schema and knowledge base of the ontology.

CHAPTER 3

ONTOLOGY-DRIVERN QUESTION ANSWERING USING SEMANTICQA

This chapter explains our approach in utilizing the schema and knowledge base of an ontology to help build and extract answer from multiple resources. The approach, named SemanticQA for Semantic Question Answering, utilizes the ontology to help the user form questions, then transforms these questions into a set of triples that the system will finally attempt to answer, mainly from the ontology, but will use one or more web documents to find answers for triples which do not have an answer in the ontology.

3.1 Introduction

Large amounts of data in many disciplines are continuously being added to semantic repositories as a result of continuing research in different scientific fields, and it is becoming an increasing challenge for researchers to use these repositories efficiently and at the same time cope with this fast pace of the introduction of new knowledge [30]. In addition to the challenges of using semantic repositories, research will naturally continue to introduce more knowledge, and it will be difficult for these semantic repositories to always contain the up-to-date knowledge that can, for example, be published in journal articles or Wikipedia pages before repositories get updated with the new knowledge.

Users therefore need an approach that allows easy access to up-to-date data from multiple resources so they can perform their duties efficiently. Increasing focus is being given currently to allowing users to type their queries in natural language instead of having to type them in a specific query language [8]. Currently, users who can belong to many fields – usually not related to computers – wishing to utilize such repositories need to know the contents of the ontology, which means understanding OWL or RDF, and know how to query these ontologies using one of the ontology query languages, e.g., SPARQL.

Such requirements are some of the major reasons the Semantic Web has not become mainstream as fewer than expected users are utilizing such knowledge [17], forming major problems for the whole Semantic Web Initiative.

SemanticQA was built to help ease the process of extracting knowledge buried inside the millions of triples ontologies have.

In summary SemanticQA has the following features:

1. Interactive question-building interface: our system allows the interaction between the user and the system to provide useful suggestions at all stages of building a question.
2. Ontology-portable system: Although our system relies on the ontology to build through all of its stages, the ontology component is a plug-in component that can be replaced with another ontology that models a completely different domain without any change in the system performance.
3. The natural language question is answered through a multi-step process that includes;
    a. Spotting ontology entities that are included in the question, taking into consideration synonyms.
    b. Using the spotted entities to form question triples.
    c. These triples are converted to a SPARQL query and run against the ontology to get answers from the ontology directly.
    d. If no answers were found in the ontology, triples are used to form keyword web searches that are used to retrieve web documents relevant to the question.
    e. Candidate answers are extracted from these documents.
    f. The candidate answers are ranked based on our novel measure, the Semantic Answer Score.

3.2 Architecture

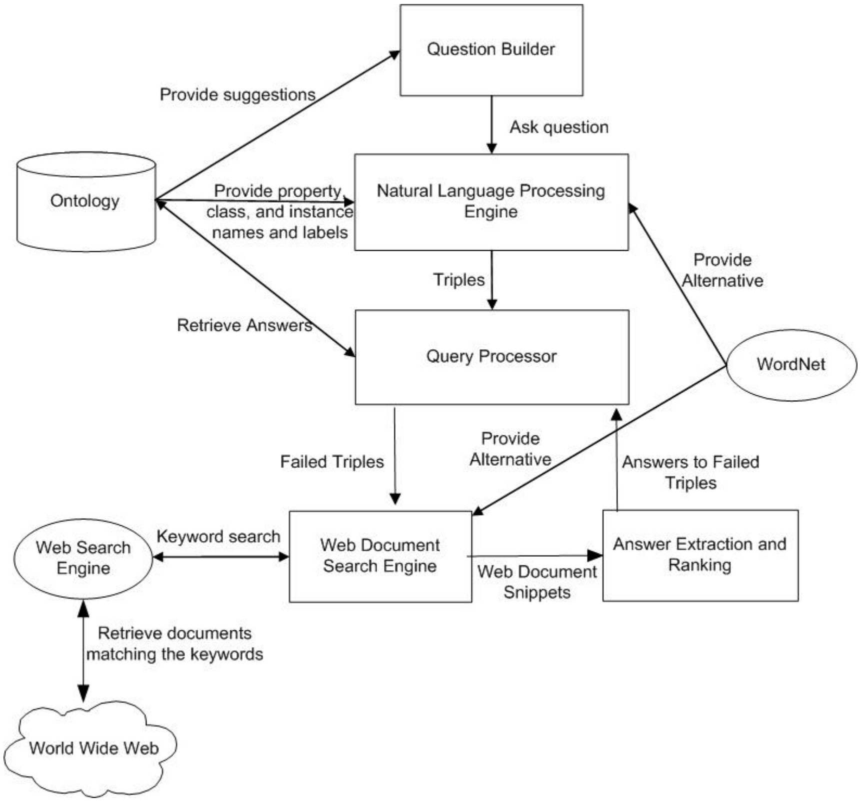


Figure 2: Architecture of SemanticQA

SemanticQA utilizes various types of knowledge in its different components to answer user questions. The question builder assists users in building their questions by utilizing the terms defined in the ontology to make suggestions based on previous user input. At the Natural Language Processing engine the question terms the user enters manually or by selecting from the suggestions are matched to ontology terms. The matches are processed and the NLP engine produces a set of query triples. These triples are sent to the Query Processor that generates a SPARQL query using the triples and runs it against the ontology to attempt to answer it from existing knowledge. If the execution of the whole query fails, the Query Engine iterates through the triples and executes each one of them separately to identify the failing triple(s) which are sent to the Web Search Engine to search for relevant web documents that are passed to the Answer Extraction and Ranking component that extracts answers from snippets of web documents.

In the next section we provide a detailed explanation of each of these components.

3.3 SemanticQA Components

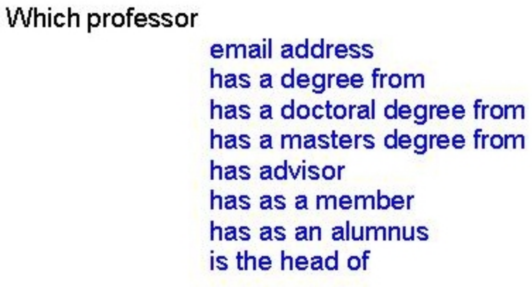
3.3.1 Ontology Example

Throughout the following subsections, a small ontology schema that represents a university scenario that was developed by Lehigh University will be used. The ontology contains information about a university domain such as professors and their levels and relationships to students and where they got their degrees from.

3.3.2 Interactive Ontology-Driven Question Builder

This component helps users who do not have enough knowledge to build complex queries corresponding to their problems in a query language (e.g., SPARQL) by allowing them to present their query in natural English uses ontology terms that are presented to the user as the query is being formed.

Depending on what the user has previously entered and what s/he has started typing, suggestions can be question words (e.g., "who", "where"), "stop" words (e.g., "the", "of", "is"), ontology class names (e.g., "professor", "student"), ontology relationship (property) names (e.g., "works for", "advisor"), or ontology instances (e.g., "John Smith", "Stanford"). For example, if the user has entered “professor”, they would be presented with suggestions based on the properties of the class “professor” in the ontology, such as “teaching”, or “advisor”, in addition to English rule words.



Which professor

Figure 3: Suggestions during the question building process

3.3.3 Natural Language Processing (NLP) Engine

This component forms the backbone of SemanticQA as its main task is to map the contents of the NL question that was built in the previous component to ontology entities prior to attempting to answer it. To identify ontology entities in the question, the Stanford Parser [33] was initially used to generate a parse tree for the question where each node in the tree represents a part of the question, that was then matched to ontology entities. The Stanford Parser was later abandoned as it was intolerant with user typing errors and was found to have difficulties in generating correct groupings for multi-word entities in these situations which are common in real-world applications. Considering that user questions are not usually too long, the processing using the Stanford Parser was replaced by the simple process of trying to match all word subsequences in the question, starting from the largest, to ontology entities.

For example, if the user asks: “Who is the advisor of Bobby McKnight?” the following subsequences are generated:

“Who is the advisor of Bobby McKnight”

“Who is the advisor of Bobby”

“is the advisor of Bobby McKnight”

“Who is the advisor of”

“is the advisor of Bobby”

“the advisor of Bobby McKnight”

“Who is the advisor”

“is the advisor of”

“the advisor of Bobby”

“advisor of Bobby McKnight”

“Who is the”

“is the advisor”

“the advisor of”

“advisor of Bobby”

“of Bobby McKnight”

“Who is”

“is the”

“the advisor”

“advisor of”

“of Bobby”

“Bobby McKnight”

“Who”

“Is”

“the”

“advisor”

“of”

“Bobby”

“McKnight”

Although this approach seems too simple, it was found to perform the matching process well considering that users may have some misspellings in their questions that can cause the Stanford Parser to generate wrong combinations of linguistic components.

SemanticQA’s matching performance is enhanced by "indexing" the data in advance (as it arrives) - an appropriate vector is built for each property, class and instance, and stored in a vector-space database constructed using Lucene [29].

The matching of each of these subsequences is performed in three phases. In the first phase, we map question word subsequences to properties in the ontologies. If a match was found then a new triple is created and is populated with the property and this triple is passed to the second phase. If a match is not found then we find alternatives to the word combination from WordNet. For each synonym we do the same process we did with the original keyword until we find a match. If no matches are found for the combination, it is forwarded to the second phase.

For example, if the question: “Who is the advisor of Bobby McKnight?” was processed against the ontology of Section 3.1, the following triple will be generated as a result of matching the question word “advisor” to the label of the ontology property “advisor”:

<null> univ-bench:advisor <null>

The second phase starts after all possible matches between question words and properties are found. In the second phase we try to complete triples passed from the first phase by mapping question words to ontology classes. If a match was found then the class is placed in the first triple that where the found match is either the domain or range of the property of that triple. If a match between a question word combination and ontology classes is not found then we find alternatives to the class label from WordNet.

In addition to the relationship triple, a new triple is added to indicate that this match is a class. For example, if the previous question was worded “What is the name of the professor who is the advisor of Bobby McKnight?” the result triples after passing the second phase will be:

?prof rdf:type uni:Professor
<null> univ-bench:advisor ?prof

Finally, we try to complete triples passed from the previous phases by mapping question words to ontology instances. If we find a match then, as we did with matched classes, the instance is placed in the first triple that where the found match is either the domain or range of the property of that triple.

So, for the previous question fact that the question words “Bobby McKnight” were matched to the label of the instance “BobbyMcKnight” will be reflected on the triple that was generated and it will become:

<univ-bench:BobbyMcKnight univ-bench:advisor null>

The <null> value for the object of the triple indicates that no match was found, which indicates that this is the answer the user is looking for.

3.3.4 Query Processor

The query processor’s task is providing the user with the best answer to the question from the ontology and web documents, if needed. The query processor first combines the triples generated in the NLP engine into a single SPARQL query.

Triples are connected to each other by finding which triple’s domain matches another triple’s range, in addition to the location of the triple. So, if the question above was changed to: “Where did the advisor of Bobby McKnight get his degree from?” the following SPARQL query will be generated:

```
SELECT ?placeLabel
WHERE {
  univ-bench:BobbyMcKnight univ-bench:advisor    ?person .
  ?person                  univ-bench:degreeFrom ?place .
  ?place                   rdfs:label            ?placeLabel .
}
```

This query is issued against the ontology to attempt to retrieve the answer directly from ontology if it exists there. If an answer was found, it is presented to the user and the execution halts. If the whole query fails, indicating that some of the triples do not have answers in the ontology, the query processor tries to identify the triple that caused the failure by going through all the triples generated in the previous step one at a time and generates a SPARQL query for that triple only and tries to execute it against the ontology. If no answer was found in the ontology for that triple, the query processor attempts to answer this triple from the web by invoking the document web search engine, and if there are more unanswered triples, web answers to the current triples are matched to ontology instances and the first match will be identified as the answer and passed to the next triple. If this is the last triple, a predetermined number of web answers (e.g., the first ten) are displayed to the user.

3.3.5 Document Search Engine (DSE)

The task of the DSE is to use the classes, relations, and entities that are used in unanswered triples in addition to the answers of previously answered triples to generate multiple keyword sets that will be sent to the web search engine to find web documents that may contain the answer(s) to the question.

The first keyword set the DSE generates is obtained using the instances and the labels of properties and classes included in the triple, in addition to the labels of the classes (types) of the question instances and the label of the expected class of the answer we are looking for as extracted from the triple.

This set is generated by first adding instances, classes and relations that were mentioned in the triple, in our case: {“Bobby McKnight”, Advisor}. Then, the DSE adds classes of the instances that are mentioned in the triple, in our case, “Student” is added, since it is the type of “Bobby McKnight”. Finally, the expected type of the answer is added to the keyword set. In our example, the triple “Student → Advisor → Professor” exists in the ontology, therefore, the keyword “Professor” is added to the keyword set, indicating that we are looking for an answer of type professor, which is likely to cause the web search engine to return documents that are more relevant to the question. The result of applying this process to the question mentioned above is the following keyword set.

“Bobby McKnight”, Advisor, Student, Professor

This first keyword set is sent to the search engine to retrieve relevant documents. To enable the system to find the answer even if it is in a document that does not contain the original question terms entered by the user, but might contain some of their alternatives, the DSE generates additional keyword sets by replacing class and property names that were included in the first keyword set by their alternatives as obtained from WordNet. For example, the following keyword set is one of the additional keyword sets the system generates for the failed triple: {"Bobby McKnight", Adviser, Student, Prof}. The documents list returned for each keyword set are collected and then transferred to the next component.

This component also allows for the user to restrict the documents to be retrieved from a single domain instead of any document from any domain that can be irrelevant to the field. For example, a user in the medical research field might want to limit the search to PubMed, to guarantee more relevant results.

3.3.6 Semantic Answer Extraction and Ranking (SAER)

SAER's task is to extract possible answers to the unanswered triples of the question using the documents the DSE retrieved from the web, and then rank these answers. The SAER utilizes the snippets of web documents that are generated by web search engines to indicate where the search terms are located in the document. A snippet of the web document is a combination of a few short sections that contain the search terms. If these sections come from separate locations of the document, web search engines usually indicate this using a special kind of separator (for example Google[8] and Yahoo[9] use "…"). We utilize these snippets and the separators to limit our document processing to only the relevant portions of the documents as determined by the search engine.

In SAER, noun phrases (According to Merriam-Webster dictionary, A noun phrase is a phrase formed by a noun and all its modifiers and determiners) within these snippets are identified by the Stanford Parser as candidate answers to the triple that was not answered from the ontology alone. Each noun phrase (NP) is given a score that we call the *Semantic Answer Score* to determine their relevance to the triple using the following formula.

[8] http://www.google.com

$$\text{Score} = W_{\text{Answer Typ}}{*}\text{Distance}_{\text{Answer Type}} + W_{\text{Property}}{*}\text{Distance}_{\text{Property}} + W_{\text{Others}}{*}\text{Distance}_{\text{Others}}$$

The score is a weighted sum of three different groups of measurements that are explained below. The measurement weights were calibrated based on empirical trials. Please note that when referring to the name of class or a property, we also refer to any of its alternatives as determined by WordNet.

1. **Distance$_{\text{Answer Type}}$**: During our experiments, we found that if the NP is very close to the expected type (class) of the answer that was a very good indication the NP is a candidate for being an answer for the unanswered triple. For example, if the search was for a "Professor", and the NP was close to the word "Professor", there is a good chance this is the answer. We utilize this observation into the score of the NP by computing this distance as the number of characters that separate the NP from the expected type of the answer in the snippet, and we penalize the NP if they were separated by "…", to indicate that there is a large distance between the NP and expected answer type.
2. **Distance$_{\text{Property}}$**: In a similar fashion, the distance that separates a NP and the property that was used in the triple we are answering determines the relevance of that NP to the triple, and we also take this fact into account by computing this distance as the number of characters that separate the NP from the property name in the snippet, and we penalize they were separated by "…".
3. **Distance$_{\text{Others}}$**: Finally, the distance that separates the NP from all other terms in the keyword set such as the named entities that were mentioned in the question or their types. The score of the NP is penalized if they are separated by "…".

This score utilizes semantic knowledge from the ontology to capture the most-likely answer to a question when extracted from web documents. For example it was noticed that NPs close to the property label of the question (e.g., Advisor) are more likely to be answers of the question than other words. As an

[9] http://www.yahoo.com

example, when the alternative keyword set {“Bobby McKnight” "Major Professor" Student Professor} is sent to the search engine, one of the document snippets includes the following:

> *“The Homepage of Bobby McKnight ... Georgia under the direction of Dr. Budak Arpinar (Major Professor), Dr. John Miller (Committee Member), and Dr. Liming ...”*

The correct answer for the query is Dr. Budak Arpinar, and as it can be seen, this answer is closer to the property name. This has been observed with most other situations as well. For example, when a question such as “Where is the U.S. Mint headquarter?”, the correct answer “Washington DC is adjacent to the word “headquarters”.

Following the property in scoring an answer is the expected answer type that was extracted from the ontology. In our advisor example, the word “professor” or one of its alternatives will probably be adjacent to the candidate answer in the retrieved document, therefore it is given the second highest weight when computing the score of a candidate answer.

The last component in the calculation of the score of a noun phrase is its distance of other question components, such as named entities (“Bobby McKnight”), or question words that do not match ontology concepts, relationships, or instances.

After all noun phrases from all snippets are extracted and scored, they are then matched against ontology instances that have a type similar to the type of the answer we are looking for, starting from NP with the highest score. This process allows for the continuation of the process of answering a question if only a part of the query (a triple) does not have answers in the ontology, but others do. For example, if the question was changed to “Where did the advisor of Bobby McKnight get his degree from?” and the advisor information does not exist in the ontology, but was extracted from web documents, the answer is matched against professors in the ontology, and if there is a match, the next task would be to extract where the advisor got his degree. This knowledge might exist in the ontology, and in this case it will presented directly to the user, otherwise, new web search and answer extraction processes start in a manner similar to the process that retrieved the advisor name.

As will be shown in Chapter 5, this process of utilizing semantic knowledge in the form of ontologies for the purpose of question answering has been successful in answering different types of questions, and with some enhancements and proper quality ontologies, can be a good tool for users in the general domain.

CHAPTER 4

ONTOLOGY EVALUATION AND RANKING USING ONTOQA

Ontologies form the cornerstone of the Semantic Web and are intended to help researchers to analyze and share knowledge, and as more ontologies are being introduced, it is difficult for users to find good ontologies related to their work. Therefore, tools for evaluating and ranking the ontologies are needed. In this dissertation, we present OntoQA, a tool that evaluates ontologies related to a certain set of terms and then ranks them according a set of metrics that captures different aspects of ontologies. Since there are no global criteria defining how a good ontology should be, OntoQA allows users to tune the ranking towards certain features of ontologies to suit the need of their applications. We also show the effectiveness of OntoQA in ranking ontologies by comparing its results to the ranking of other comparable approaches as well as expert users.

4.1. Introduction

The Semantic Web envisions making the content of the web processable by computers as well as humans. This is mainly accomplished by the use of ontologies which contain terms and relationships between these terms that have been agreed upon by members of a certain domain (e.g., the Gene Ontology (GO)[10] and other ontologies in biology such as the Open Biology Ontologies (OBO), or academia such as SWETO-DBLP [2], as well as general-purpose ontologies like TAP in [24]. These agreed upon ontologies can then be published to be available for use by other members of the domain.

Building ontologies can be accomplished in one of two approaches: it can start from scratch [14], or it can be built on top of an existing ontology [14]. In both cases, techniques for evaluating the resulting ontology are necessary [18]. Such techniques would not only be useful during the ontology engineering

[10] http://www.geneontology.org

process [46], they can also be useful to an end-user who needs to find the most suitable ontology among a set of ontologies.

These techniques will be particularly useful in domains where large ontologies including tens of classes and tens of thousands of instances are common. For example, a researcher in the bioinformatics domain who is looking for an ontology that is mainly concerned with genes might have access to many ontologies (e.g., MGED[11], GO, OBO) that cover very similar areas, making it difficult to simply glance through these ontologies to determine the most suitable ontology. In such situations, a tool that would provide an insight into the ontology and describe its features in a way that will allow such a researcher to make a well-informed decision on which ontology to use will be helpful.

OntoQA is a suite metrics that evaluate the content of ontologies through the analysis of their schemas and instances in different aspects such as the distribution of classes on the inheritance tree of the schema, the distribution of class instances, and the connectivity between instances of different classes. In addition, OntoQA utilizes this set of metrics to rank ontologies related to a user supplied set of terms.

It is important to highlight that ontology features largely depend on the domain the ontology is modeling, therefore, OntoQA allows users to bias the ranking so ontologies that possess certain characteristics (e.g., ontologies with inheritance-only relationships, or deep ontologies) are ranked higher.

Thus, our contributions in this part of the dissertation can be summarized as the following:

- A flexible technique to rank ontologies based on their contents and their relevance to a set of keywords as well as user preferences.
- According to our knowledge OntoQA is the first approach that evaluates ontologies using their instances (i.e. populated ontologies) as well as schemas.

[11] http://mged.sourceforge.net/ontologies

4.2. Architecture

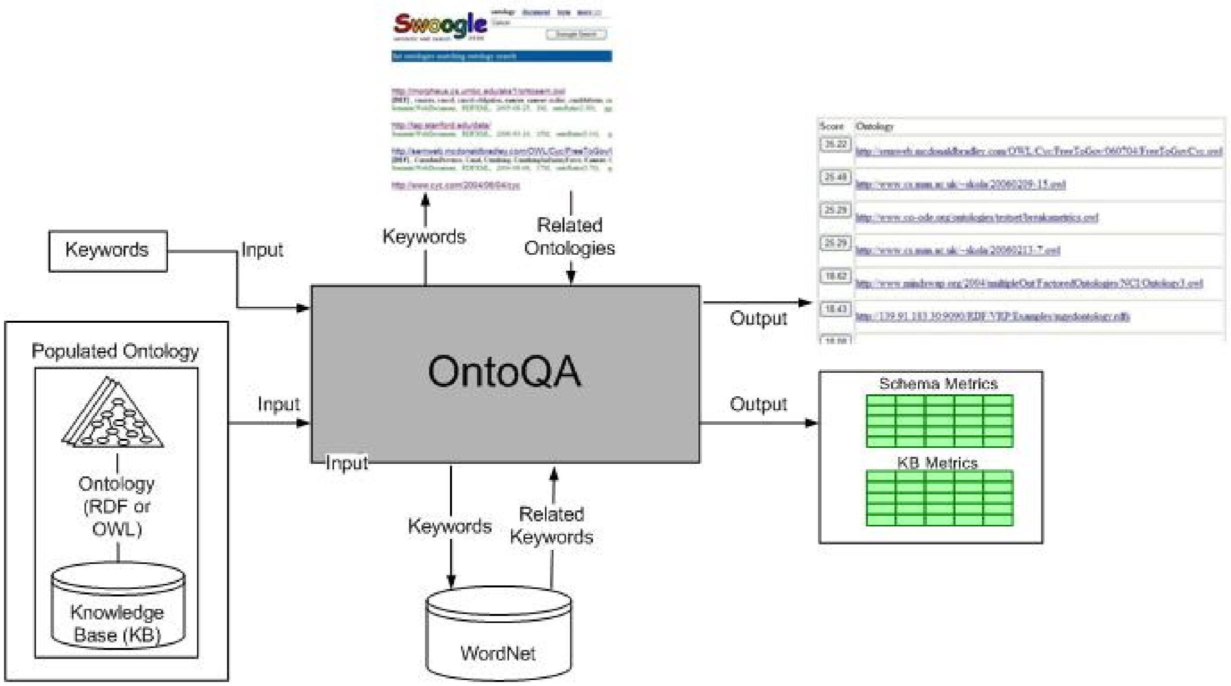


Figure 4: Architecture of OntoQA

OntoQA was implemented as a public Java web application[12] that uses Sesame [10] as an RDF repository, Figure 4 shows the overall structure of OntoQA. Depending on the input, there are three scenarios for using OntoQA. Here is a step-by-step explanation of how the different OntoQA components are utilized in each case:

1. Ontology:
    a. OntoQA calculates metric values.
2. Ontology and keywords:
    a. OntoQA calculates metric values.
    b. OntoQA uses WordNet to expand the keywords to include any related keywords that might exist in the ontology.

[12] http://128.192.251.199:8000/OntoQA/

c. OntoQA uses the metric values to obtain a numeric value that evaluates the overall contents of the ontology and its relevance to the keywords.

3. Keywords:

a. OntoQA uses Swoogle to find the RDF and OWL ontologies in the top 20 results returned by Swoogle.

b. OntoQA then evaluates each of the ontologies as indicated in case 2 above.

c. OntoQA finally displays the list of ontologies ranked by their score.

4.3. Terminology

In this section we highlight the main elements of the terminology. The schema of an ontology consists of the following main elements:

- A set of classes, C.
- A set of relationships, P.
- An inheritance function, H^C.
- A set of class attributes, Att.

The knowledgebase of an ontology consists of the following main elements:

- A set of instances, I.
- A class instantiation function, inst(C_i).
- A relationship instantiation function, instr(I_i, I_i).

In addition to the above terms, we introduce the following terms that will be used in the following section:

- The set of class-ancestor pairs in the ontology, $H := \{(C_i, C_j)$, where $i \neq j\}$.
- The set of class-ancestor pairs in the inheritance subtree rooted at C_i: $H(C_i) := \{(C_j, C_i)$, where $i \neq j$ and $H^C(C_j, C_i)\}$
- The set of subclasses of a class C_i: $SubCls(C_i) = \{C_j$, where $H^C(C_j,C_i)\}$.
- The set of relationships a class C_i has with another class C_j: $CREL(C_i) := \{ \cup P(C_i, C_j)\}$.

- The set of distinct relationships used by instances of a class C_i: $IREL(C_i) := \{ \cup instr(I_i, I_j)$, where $I_i \in inst(C_i)\}$.

- The number of all relationships used by instances of a class C_i: $SIREL(C_i) := \{\sum|instr(I_i, I_j)|$, where $I_i \in inst(C_i)\}$..

- The set of non-empty classes in the ontology: $C' := \{C_i$, where $inst(C_i) \neq Ø\}$.

- The number of instances of a class C_i as expected by the user: $Expected(C_i)$.

4.4. OntoQA Metrics

We divide the evaluation of an ontology on two dimensions: Schema and instances. The first dimension evaluates ontology design and its potential for rich knowledge representation. The second dimension evaluates the placement of instance data within the ontology according to the knowledge modeled in the schema.

In the following sections we will define metrics to evaluate each of the above dimensions. These metrics are intended to evaluate certain aspects of ontologies and their potential for knowledge representation.

4.4.1. Schema Metrics

The schema metrics address the design of the ontology schema. Although it is difficult to know if the ontology design correctly models the knowledge of the domain it is trying to represent, we provide some metrics that indicate different features of an ontology schema.

Relationship Diversity: This metric reflects the diversity of relationships in the ontology. An ontology that contains mostly inheritance relationships (taxonomy) usually conveys less information than an ontology that contains a diverse set of relationships. However, in some applications, users might be interested in ontologies with mostly inheritance relationships (e.g., species classification), and OntoQA gives the user the option to specify whether she prefers a taxonomy or an ontology with diverse relationships.

Definition 1: The relationship diversity (RD) of a schema is defined as the ratio of the number of non-inheritance relationships (P), divided by the total number of relationships defined in the schema (the sum of the number of inheritance relationships (H) and non-inheritance relationships (P)).

$$RD = \frac{|P|}{|H| + |P|}$$

For example, if an ontology has an RD value close to 0 that would indicate that most of the relationships are inheritance relationships. In contrast, an ontology with a value close to 1 would indicate that most of the relationships are non-inheritance.

Schema Deepness: This measure describes the distribution of classes across different levels of the ontology inheritance tree. This measure can distinguish a shallow ontology from a deep ontology. A shallow ontology is an ontology that has a small number of inheritance levels, and each class has a relatively large number of subclasses. In contrast, a deep ontology contains a large number of inheritance levels where classes have a small number of subclasses

Definition 2: The schema depth of the schema (SD) is defined as the average number of subclasses per class.

$$SD = \frac{|H|}{|C|}$$

An ontology with a low SD would be deep, which indicates that the ontology covers a specific domain in a detailed manner (e.g., ProPreO [49]), while an ontology with a high SD would be a shallow (or horizontal) ontology (e.g., TAP), which indicates that the ontology represents a wide range of general knowledge with a low level of detail.

4.4.2. Instance Metrics

The way instances are placed within an ontology is also a very important aspect of ontology evaluation. The placement of instance data and distribution of the data can indicate the effectiveness of the ontology design and the amount of knowledge represented by the ontology. Instance metrics can divided on three main sub-dimensions: Overall KB (knowledgebase) metrics that evaluates the overall placement of instances with regard to the schema, class-specific metrics that evaluate the instances of a specific class and compare it to instances of other classes, and relationship-specific metrics that evaluate the instances of a specific relationship and compare it to instances of other relationships.

4.4.2.1 Overall KB Metrics

This group of metrics gives an overall view on how instances are represented in the KB.

Class Utilization: This metric reflects how classes defined in the schema are being utilized in the KB. This metric can be used to differentiate between two ontologies having the same classes defined in their schemas but one of them populates more classes than the other one, indicating a richer KB.

Definition 3: The class utilization (CU) of an ontology is defined as the ratio of the number of populated classes (C`) divided by the total number of classes defined in the ontology schema (C).

$$CU = \frac{|C`|}{|C|}$$

The result will be a percentage indicating how the KB utilizes classes defined in the schema. Thus, if the KB has a very low CU, then the KB does not have data that exemplifies all the knowledge that exists in the schema. This metric will be very useful in situations where instances are being extracted into an ontology and it is needed to evaluate the results of the extraction process.

Cohesion: This metric represents the number of connected components in the KB. This metric can particularly help if “islands” form in the KB as a result of extracting data from separate sources that do not have common knowledge, giving insight into what areas need more instances in order to enable the different connected components to connect to each other. Having less connected components (ideally 1) can be helpful, for example, in finding more useful semantic-associations [4] in the ontology.

Definition 4: The cohesion (Coh) of an ontology is defined as the number of connected components (CC) of the graph representing the KB.

$$Coh = |CC|$$

The result will be an integer representing the number of connected components in the ontology.

Class Instance Distribution: This metric is also useful to evaluate the instance extraction process. It provides an indication on how instances are spread across the classes of the schema. It can be used to discover problems in the instance extraction process.

Definition 5: The class instance distribution of an ontology is defined as the standard deviation in the number of instances per class.

$$\text{CID} = \text{StdDev}(\text{Inst}(C_i))$$

4.4.2.2 Class-Specific Metrics

This group of metrics indicates how each class defined in the ontology schema is being utilized in the KB.

Class Connectivity: This metric gives an indication of the centrality of a class. With the importance metric mentioned below, both metrics provide a better understanding of how focal some classes are in the KB, which might be help in cases where a user has two ontologies with the similar classes defined in their schemas, but classes that are be important to the user play a central role in one of them, while being on the boundary in the other.

Definition 6: The connectivity of a class (Conn(C_i)) is defined as the total number of relationships instances of the class have with instances of other classes.

$$Conn(C_i) = \left|NIREL(C_i)\right|$$

Class Importance: This metric is important because it helps in identifying which areas of the schema are in focus when the instances are extracted and inform the user of the suitability of his/her intended use. It will also help direct the ontology developer or data extractor to where s/he should focus on getting data if the intention is to get a consistent coverage of all classes in the schema. Although this measure does not consider the real world semantics, where some classes naturally have more instances than others, the class importance can still be used (together with the class connectivity measure mentioned above) to give an indication on what parts of the ontology are considered focal and what parts are on the edges.

Definition 7: The importance of a class (Imp(C_i)) is defined as the number of instances that belong to the inheritance subtree rooted at C_i in the KB (inst(C_i)) compared to the total number of class instances in the KB (CI).

$$Imp(C_i) = \frac{|Inst(C_i)|}{|KB(CI)|}$$

Relationship Utilization: This metric reflects how the relationships defined for each class in the schema are being used at the instances level. It is a good indication of the how well the extraction process performed in the utilization of information defined at the schema level. This metric can be used to distinguish between two ontologies having similar schemas but one of them utilizes only a few of the available relationships while other utilizes more.

Definition 8: The relationship richness (RU) of a class C_i is defined as the number of relationships that are being used by instances I_i that belong to C_i ($P(I_i,I_j)$) compared to the number of relationships that are defined for C_i at the schema level ($P(C_i,C_j)$).

$$RU(C_i) = \frac{|IREL(C_i)|}{|CREL(C_i)|}$$

4.4.2.3 Relationship-Specific Metrics

This group of metrics indicates how each relationship defined in the ontology schema is being utilized in the KB.

Relationship Importance: This metric measures the percentage of instances of a relationship with respect to the total number of relationship instances in the KB. This metric is important in that it will help in identifying which schema relationships were in focus when the instances were extracted and inform the user of the suitability of his/her intended use. This metric can also help in directing the instance extraction process to include a more diverse set of relationships the KB does not include the required diversity.

Definition 9: The importance of a relationship (Imp(R_i)) is defined as the number of instances of relationship R_i in the KB (inst(R_i)) compared to the total number of property instances in the KB (RI).

$$Imp(R_i) = \frac{|Inst(R_i)|}{|KB(RI)|}$$

The result of the formula will be a percentage representing the importance of the current class.

4.5. Ontology Score Calculation

If the user is searching for ontologies related to a set of terms or is trying to evaluate an ontology regarding a set of terms, OntoQA evaluates the ontology based on the entered keywords in the following manner:

1. The terms entered by the user are extended by adding any related terms obtained using WordNet.
2. OntoQA determines the classes and relationships whose names contain any term of the extended set of terms.
3. OntoQA finally aggregates the schema, the overall KB metrics, and the metrics for all the related classes and relationships to get an overall score for the ontology.

Definition 15: The score of an ontology can be measured as the weighted average of schema metrics, overall KB metrics, and the metrics of related classes and relationships.

$$Score = \sum W_i * Metric_i$$

Where:

Metric{} = {RD, SD, CU, Coh, #Classes, #Relationships, #Instances, Avg(Conn(C_i)), Avg(Imp(C_i)), Avg(RU(C_i)), Avg(Imp(R_i))} is the set of metrics used in calculating the overall score of an ontology (the averages are for classes and relationships related to the keywords).

W{} is the set of weights for each metric.

Please note that the initial values for the set of weights W was set based on empirical testing and can be adjusted with more testing. Also, these weights can be modified by the user to reflect his favor of certain aspects of the ontology.

Among the metrics used to compute the overall score, the relationship diversity (inheritance vs. diverse relationships) and the class deepness (shallow vs. deep ontologies) can be biased towards either option based on the user preference. The other metrics such as the class utilization, connectivity, and importance metrics are always preferred to be higher in better ontologies.

The overall score reflects the overall nature of the ontology and how much it relates to the keywords.

CHAPTER 5

EXPERIMENTAL EVALUATION

We have implemented our approaches for question answering and ontology evaluation. In this chapter we provide implementation details are the results of the evaluation of these implementations.

5.1 Question Answering using SemanticQA

SemanticQA was implemented as a Java application with a web interface. The implementation was named SemanticQA, for Semantic Question Answering. The implementation uses the open source Semantic Web framework Jena [12] to access ontologies and Google to search for web pages. For accessing ontologies of large size which can get to the millions of triples (e.g., DBPedia contains about seven million triples), the default memory-based model in Jena fails to even load such ontologies. The alternative to this approach was to use a persistent store model for storing ontologies. The persistent model uses a relational database for storing ontology triples. For the relational database, the open source MySQL[13] RDBMS was used. This approach allows the implementation to load an ontology once as a preparation phase prior, and then use it to answer questions later without the need to load the ontology again. For example, when DBPedia's "infobox" ontology was loaded to the database on a Dell Dimension desktop with a Pentium 4 3.00 GHz processor with 2GB of memory running Windows XP, the process of loading the ontology to the database took approximately 6 hours.

Since matching question words to ontology entities is a string matching process that has to go through all instances for example to find an instance match to user terms, this process needed to be designed carefully to handle large ontologies. The implementation of choice was to utilize Java's HashMaps to store labels of ontology instances to utilize Java's hashing techniques instead of having to loop through all triples manually to attempt to match an instance to a user term. Prior to each run of the

[13] http://www.mysql.com

system, the instance labels are loaded into a memory-based HashMap that is later used in the matching process. In the DBPedia example, this process took approximately 5 minutes on the same machine described above. A slight difference in implementation using a Java ArrayList instead of the HashMap caused the matching process to take 20 minutes instead of one or two seconds in the HashMap case.

In addition to DBPedia, other large and small ontologies were tested against SemanticQA. Results of these tests are presented below. To have a more comprehensive test, the system is being tested now using the DBPedia ontology to answer TREC question answering track questions. TREC (Text Retrieval Conference) is an on-going series of workshops focusing on a list of different information retrieval (IR) tracks that is sponsored by the National Institute of Standards and Technology.

TREC QA track questions of last workshop include 445 questions about 70 topics ranging from people, to organizations, to events and others types. Out of these 445 questions, 360 questions are factoid questions, asking about a single fact (e.g., Who was the judge in the Robert Blake criminal trial?), and the rest are list questions, where the answer include multiple entities (e.g., Name people who have appeared on the U.S. Mint's coins.)

Results of this test will be available in the coming few days. TREC QA track tests include a certain set of news documents that track participants need to use to extract answers. But since our system is a using the web as a “document set”, and since TREC’s questions are general-purpose about well known entities (e.g., US Mint, Paul Krugman) that are well-represented in Wikipedia[14] articles, tests will be limiting the retrieved documents to be only from Wikipedia.

Below we evaluate the effectiveness of the system in answering factoid questions in multiple domains using three published ontologies to the system and comparing the answers retrieved to the correct answers of these questions, and presenting the results as instance recall (IR) and instance precision (IP). We use the definitions of IR and IP as defined in [36]:

Definition: Let:

[14] http://www.wikipedia.org

S: the number of known correct answers to a question,

D: the number of correct results found by the system,

N: the number of total results found by the system.

Then, IP and IR can be defined as:

IP = D/N, IR = D/S.

5.1.1 SwetoDblp

SwetoDblp is a widely-used publically-available large-scale populated ontology that models on computer science publications such as: proceedings, book chapters, conferences, author, affiliations, and co-editorships. We asked the system six sample questions that covered different classes and properties in the ontology.

Of the six questions asked, the system was able to find the correct answers (in the top five results) five times (IR = 83%) and the correct answer was the first answer in all those times it found the answer five times (IP = 83%).

A sample of the questions is shown in Table 3. As can be seen in the table, the only time the system could not find an answer was in the case of the question "What university is Amit Sheth at?" and the system returned the answer of "University of Georgia" as the top result, which is not totally wrong, since Amit Sheth was working there until very recently, resulting in many hits in Google.

Table 3: Sample Questions and Answers using SwetoDblp

Question	Correct Answer	Rank
What is the volume of A Tourists Guide through Treewidth in Acta Cybernetica?	Volume 11	1
What is the journal name of A Tourists Guide through Treewidth?	Acta Cybernetica	1
What university is Amit Sheth at?	Wright State University*	Not Found
What is the ISBN of Database System the Complete Book?	ISBN-10: 0130319953	1

* Amit Sheth recently moved to this university, and Google's indices might have not captured this fact yet.

5.1.2 Lehigh University Benchmark

The Lehigh University ontology is widely-used benchmark that describes terms relating a university domain. This ontology contains various types of information about the academic domain such as professors, advisors, and departments. The ontology was fitted with a small factual dataset that represents several professors, students, and institutions. Eight factoid questions using various properties and classes in that ontology were asked to the system and the answers retrieved were compared to the known correct answers.

For each question the system was able to find the correct answer to the question resulting in an IR score of 100%. The IP score for this ontology was 63%. Table 4 provides a sampling of the questions and the expected answer(s) as well as that answer's ranking in the results. The system was able to correctly process inverse relationships and one-to-many relationships. For example, a professor can be the advisor of many students. In cases as these many correct answers are possible and the system is able to find many such correct answers. It is also worth mentioning that while Krys Kochut was the head of the Computer Science Department at UGA, using the same question to find the head of the Computer Science Department at UGA in Google as a regular user would, did not return Krys Kochut, but the system was able to find the correct answer because it uses the ontology to find the that the type of the desired answer is professor and then finds that one label of professor is "Dr.", and that keyword combination of "head Computer Science UGA Dr." causes Google to return a document that contains the correct answer.

Table 4: Sample Questions and Answers using LeHigh

Question	Correct Answer (s)	Rank
Who is the advisor of Samir Tartir?	Dr. Budak Arpinar	1
Who is Budak Arpinar the advisor of?	Boanerges Aleman-Meza Samir Tartir Bobby McKnight	2 4 Not found
Who is head of the Computer Science Department at UGA?	Eileen Kraemer*	Not found

* Krys Kochut was head of the department until last year.

5.1.3 ComGo

The system was evaluated using the ComGo [40] biological ontology. Eight different questions were asked and a correct answer was found in the top five results seven out of eight times and the correct answer was the top result five times, and was first five times. This results in an IR score of 88% and an IP score of 63%. Table 5 shows some questions and their expected answer(s) as well as the answer's ranking in the results. It was also able to correctly process inverse relationships.

Table 5: Sample Questions and Answers using ComGo

Question	Correct Answer	Rank
What is the gene id of ribosome?	GO:0005840	1
What is the gene ontology term of GO:0005840?	Ribosome	3
Which organisms are epimastigotes?	Trypanosoma cruzi	1
What are the pfam domains of AT1G13300?	PF00249	1

Combing the statistics from the three test cases results in global IR score of 90% and a global IP score of 69%, showing that our system is very successful in utilizing the different information it find in the ontology to answer user questions, and as in the case of the head of the department of computer science at UGA, it found the correct answer, although the asking Google multiple regular queries did not return the correct answers.

5.2 Ontology Evaluation Using OntoQA

5.2.1 Ontology Search and Ranking

To illustrate the effectiveness of OntoQA in ranking ontologies, we compare the ranking of the same ontologies by OntoQA, Swoogle, and a group of expert users. We also compare our results with AKTiveRank (presented in Chapter 2), which is one of the most comparable ranking approaches, using Pearson's Correlation Coefficient.

Table 6 shows the top nine RDF and OWL ontologies ranked by Swoogle when searched for the term "Paper". Each ontology is given a Roman numeral that will be used as a reference to the ontology in other figures. Note that inaccessible ontologies returned by Swoogle are eliminated from this list.

Table 6: Ontologies ranked by Swoogle

Symbol	Ontology URL
I	http://ebiquity.umbc.edu/ontology/conference.owl
II	http://kmi.open.ac.uk/semanticweb/ontologies/owl/aktive-portal-ontology-latest.owl
III	http://www.architexturez.in/+/--c--/caad.3.0.rdf.owl
IV	http://www.csd.abdn.ac.uk/~cmckenzi/playpen/rdf/akt_ontology_LITE.owl
V	http://www.mindswap.org/2002/ont/paperResults.rdf
VI	http://owl.mindswap.org/2003/ont/owlweb.rdf
VII	http://139.91.183.30:9090/RDF/VRP/Examples/SWPG.rdfs
VIII	http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl
IX	http://www.mindswap.org/2004/SSSW04/aktive-portal-ontology-latest.owl

The same term is used in OntoQA producing the results shown in Figure 5. In this figure, the contribution of each metric in the overall score is depicted with different regions in the column for each ontology, and weights are assigned to give a more balanced contribution to each metric, whereas Figure 6 presents results that are biased towards favoring larger ontology schemas.

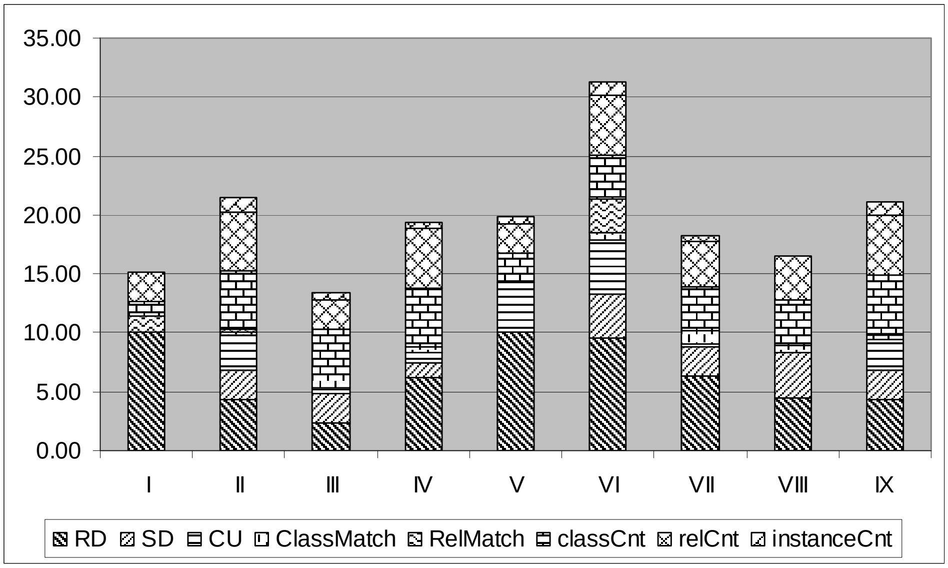


Figure 5: OntoQA results with balanced weights

In Figure 5, ontology VI is ranked the highest. This ontology has a set of 62 rich relationships between its 22 classes, an average of 3 subclasses per parent class, and almost half of the classes are populated. It also has 12 relationships that are related to papers (e.g., author, published in, abstract, etc). All these facts contribute to give this ontology the highest rank.

The differences between OntoQA's and Swoogle's rankings are obvious in the figure. The main reason for this difference is that Swoogle follows the OntoRank approach that is similar to Google's PageRank approach [44], which gives preference to "popular" ontologies. On the other hand, OntoQA ranks ontologies according to their quality measured by the different metrics tuned by users according to their preferences.

A problem with Swoogle's approach is that if the two copies of the same ontology are placed in two different locations and one of these locations is cited more than the other, Swoogle will rank the copy at this popular location higher than the other copy, even though their contents are the same, while OntoQA will give both ontologies the same ranking.

To further evaluate our approach, the same set of ontologies was ranked by two graduate students in our research lab who are not related to OntoQA and have a longtime experience in building and populating very large scale ontologies (e.g., SWETO-DBLP). These users ranked the ontologies with no relationship to a particular application, which resulted in considering ontologies with larger schemas (number of classes and relationships) as better than ontologies with smaller schemas, even if they were richer ontologies. Their ranking results are shown in Table 7.

Table 7: Ontologies ranked by users

Ontology	Rank
I	9
II	1
III	5
IV	6
V	8
VI	4
VII	2
VIII	7
IX	3

To capture their preferences we re-run our experiment after setting metric weights (W_i) so ontologies with larger schemas are ranked higher, producing the results in Figure 6. Note that other users with particular applications in mind may have different preferences than the expert users in our experiment. Therefore, OntoQA provides flexibility in allowing users with different needs to find ontologies that match their specific needs.

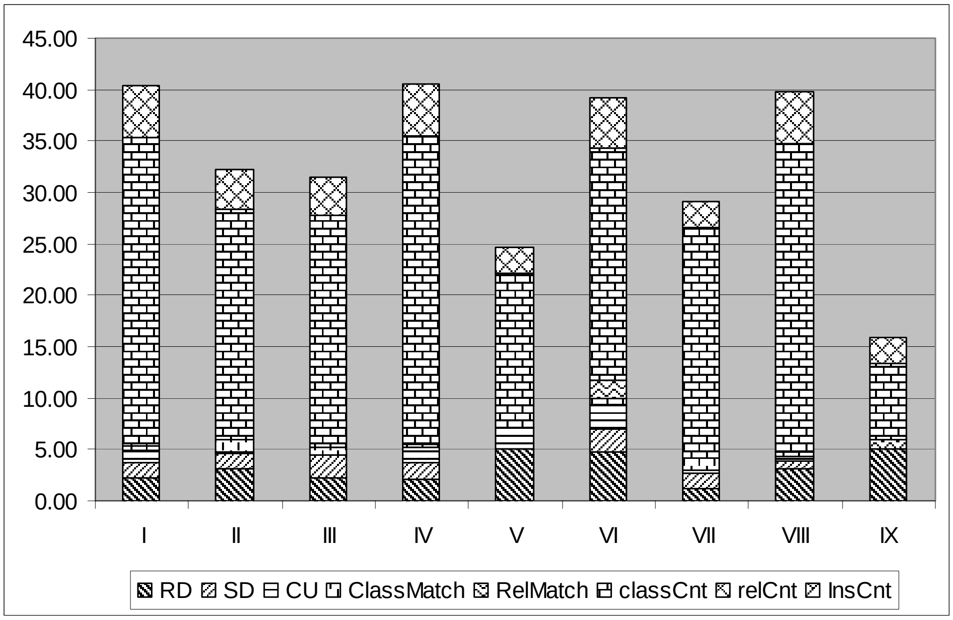


Figure 6: OntoQA results with higher weight for schema size

In this experiment, ontology IV is ranked highest due to its larger schema size (60 classes and 81 relationships). We compare the results in Figure 6 with the user ranking in Table 7. Pearson's Correlation Coefficient between the two ranking results is 0.80 indicating a relatively high correlation. When AKTiveRank is used in a similar situation [1], Pearson's Correlation Coefficient is 0.54 according to our calculation, indicating that the results of OntoQA reflect users' rankings better. We attempted to run AKTiveRank using the same term that was used here but at the time of writing this paper, it was not available publicly.

To further illustrate the usefulness of OntoQA to compare ontologies, below is a table that shows the characteristics of several ontologies, including ontologies in the OBO.

Table 8: Information about different ontologies extracted using OntoQA

Ontology	No. of Terms	Avg. No. of Subterms	Connectivity
GlycO	382	2.5	1.7
ProPreO	244	3.2	1.1
MGED	228	5.1	0.33
Biological Imaging methods	260	5.2	1
Protein-protein interaction	195	4.6	1.1
Physico-chemical process	550	2.7	1.3
BRENDA	2,222	3.3	1.2
Human disease	19,137	5.5	1
GO	200,002	4.1	1.4

Using these values, we can notice that the first two ontologies have an intermediate number of terms when compared to the rest of the OBO ontologies, which indicates that the information they contain is of an adequate size in the biological domain. The average number of sub terms per term in all ontologies is relatively similar, which also indicates that the first two ontologies have an adequate information distribution across the different levels of the term inheritance tree. It can be also seen that GlycO terms have higher connectivity to other terms in the ontology when compared with the other ontologies. This indicates that the interactions between the terms in GlycO are higher than that of the other ontologies, while the number of interactions between ProPreO terms is relatively similar to the OBO ontologies.

5.2.2 Ontology Analysis

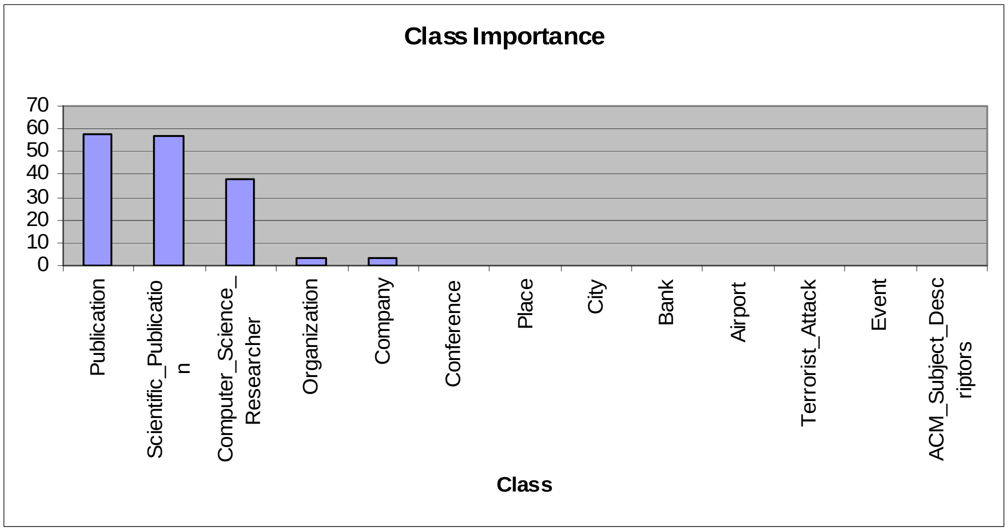
In addition to allowing users to compare ontologies, OntoQA allows users to analyze a single ontology to understand the nature of the ontology. For example, below is a table that lists summaries obtained from OntoQA about three ontologies, SWETO, TAP, and GlycO.

Table 9: Ontology summaries obtained by OntoQA

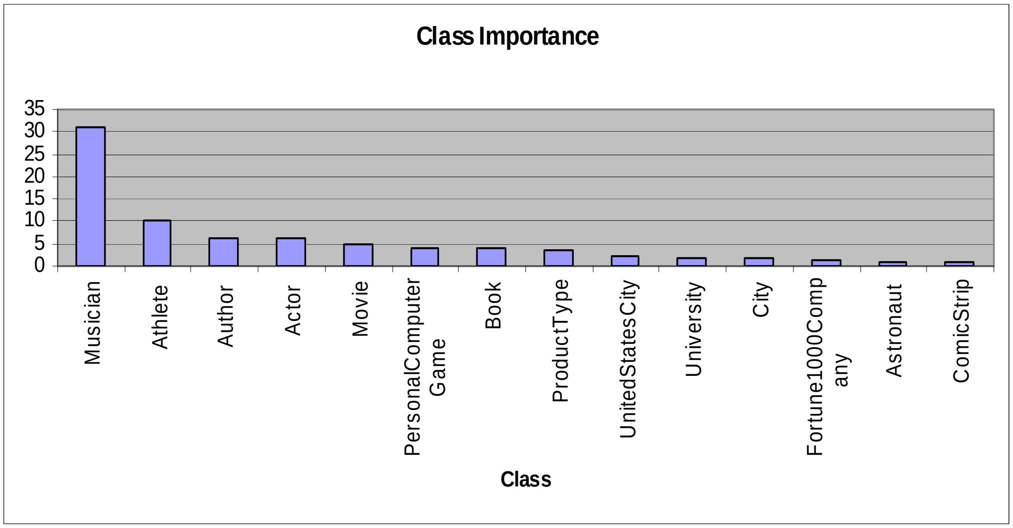
Ontology	Number of Classes	Number of Instances	Inheritance Richness
SWETO	44	813,217	4
TAP	3,229	70,850	5.36
GlycO	352	2,034	1.56

Class Importance

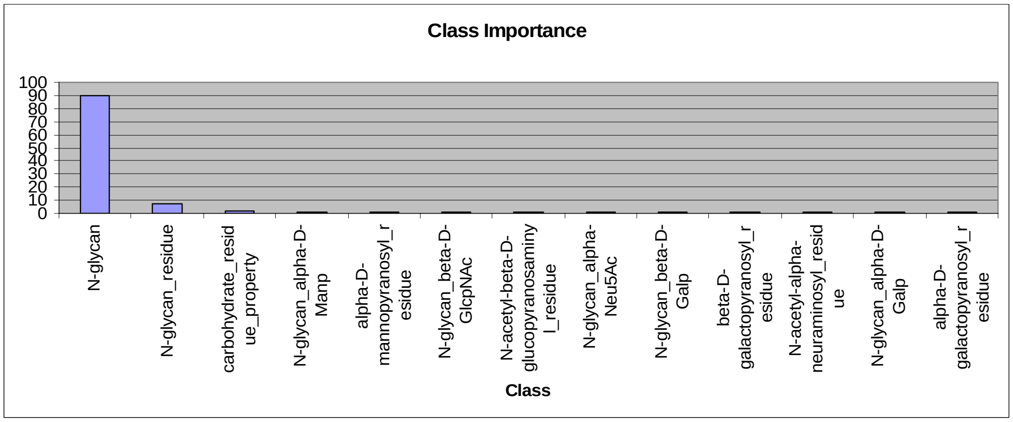
These ontologies are further analyzed to understand their nature. For example, to understand which classes in each ontology were the most important, the results in the figure below were generated by OntoQA.



(a)



(b)



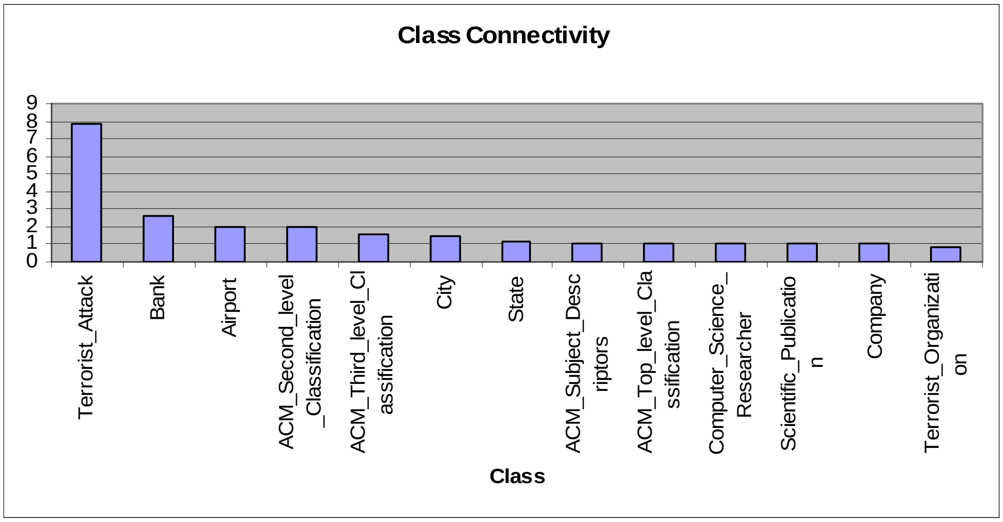
(c)

Figure 7: Class importance in (a) SWETO (b) TAP and (c) GlycO using OntoQA

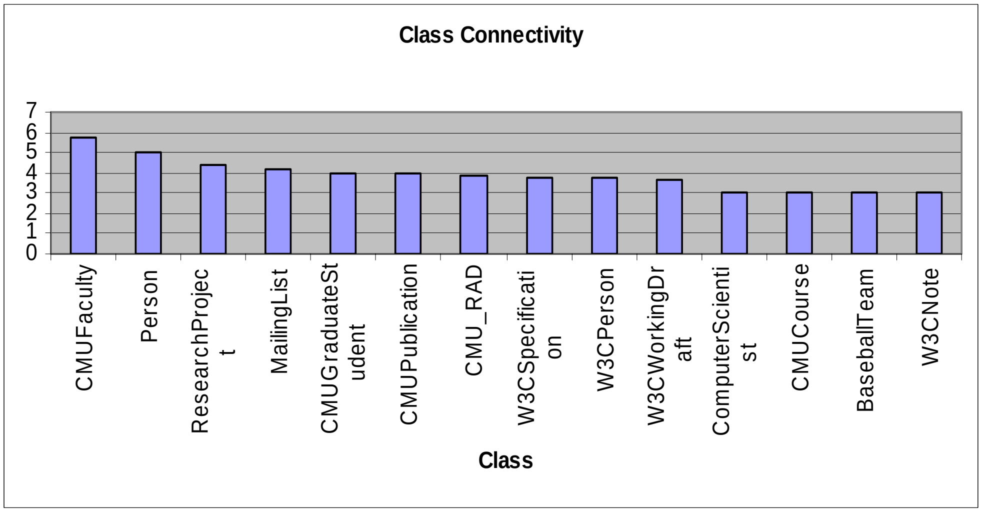
Using these results, it can be clearly seen that classes related to publications are the dominant classes in SWETO. While, with the exception of the Musician class, TAP gives consistent importance to most of its classes covering the different domains it includes. The nature of the GlycO ontology is reflected in the classes that are most important in it. The importance of the "N-glycan_residue" and the "alpha-D-mannopyranosyl_residue" and other classes show the narrow domain GlycO is intended for, although the "glycan_moiety" class is the most important class covering about 90% of the instances in the KB.

Class Connectivity

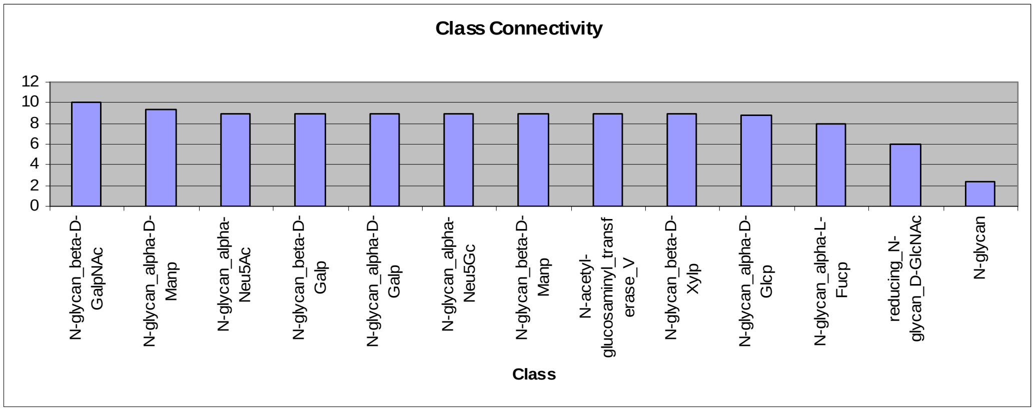
As it was discussed in Chapter 4, class connectivity is used to indicate the classes that play a more central role than other classes, which is another way of describing the nature of an ontology. Figure 8 shows the most connected classes in each of the three ontologies.



(a)



(b)



(c)

Figure 8: Class connectivity in (a) SWETO (b) TAP and (c) GlycO using OntoQA

The figures above shows that SWETO also includes good information about domains other than publications, including the terrorism domain (Terrorist_Attack and Terrorist_Organiztion), the business domain (Bank and Company) and geographic information (City and State). In a similar manner, TAP continues to show that it covers different domains as its most connected classes cover the Education domain (CMUCourse and CMUSCS_ResearchArea), the Entertainment domain (TV and Movie), and other domains as well. GlycO's specific-purpose nature is evident from the Glycan related classes that are most connected.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this dissertation, SemanticQA was introduced to combine different techniques to provide an easy-to-use interface to answer questions from multiple sources. OntoQA was also presented to provide means for users to evaluate ontologies and compare between different ontologies using a rich set of metrics and by their relation to a set of keywords.

Although SemanticQA was shown to perform well through our preliminary results, current tests on larger data sets (DBPedia) and real-world evaluation question sets (TREC) are showing promising results.

To further improve SemanticQA, we are considering processing whole web documents rather than the snippets of these web documents as produced by the web search engine. This can cause elimination of problems that are caused by truncation used by search engines to produce snippets, and will allow sentences that are better processed by English language parsers to extract possible answers. Still, this will present a challenge of text processing as web documents are frequently filled with content that is irrelevant to its contents such as advertisements and navigational panels in different sections of the document. In addition, we are working on adding the capabilities to answer more complex questions that will require query execution planning and dividing the main query into subqueries in advance to allow faster retrieval of the answers.

In regards ontology evaluation, OntoQA is shown to be different from other approaches in that it is tunable, requires minimal user involvement, and considers both the schema and the instances of a populated ontology. Our approach was well received by other researchers in the field, and in total, our work on ontology evaluation was cited more than 35 times.

To further improve OntoQA, we plan on using BRAHMS [32], or Jena instead of Sesame to handle ontologies since these two are more efficient in handling large ontologies that are common in bioinformatics. We also plan to enable the user to specify an ontology library (e.g., OBO) to limit the search in ontologies that exist in that specific library.

REFERENCES

[1] H. Alani, C. Brewster and N. Shadbolt. Ranking Ontologies with AKTiveRank. 5th International Semantic Web Conference. November, 5-9, 2006.

[2] B. Aleman-Meza, F. Hakimpour, I.B. Arpinar, and A.P. Sheth. SwetoDblp Ontology of Computer Science publications. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, (Accepted Manuscript) 2007

[3] G. Antoniou and F. van Harmelen. Web Ontology Language: OWL. In Handbook on Ontologies in Information Systems, pages 67–92, 2003.

[4] K. Anyanwu, and A.P. Sheth. ρ-Queries: Enabling Querying for Semantic Associations on the Semantic Web. In *Proc. Twelfth International World Wide Web Conference*, Budapest, Hungary, pages 690-699, 2003

[5] I.B. Arpinar, K. Giriloganathan, and B. Aleman-Meza. Ontology Quality by Detection of Conflicts in Metadata. In *Proc. Fourth International EON Workshop: Evaluation of Ontologies for the Web*, Edinburgh, Scotland, 2006

[6] S. Auer, and J. Lehmann. What have Innsbruck and Leipzig in common? Extracting Semantics from Wiki Content. In *Proc. 4th European Semantic Web Conference*, Innsbruck, Austria, pages 503-517, 2007

[7] T. Berners-Lee, J. Hendler, and O.Lassila. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 2001

[8] A. Bernstein, E. Kaufmann, A. Gohring, C. Kiefer. Querying Ontologies: A Controlled English Interface for End-Users. In: International Semantic Web Conference. (2005) 112-126

[9] H. Boley, S. Tabet and G. Wagner. Design Rationale of RuleML: A Markup Language for Semantic Web Rules. In the first Semantic Web Working Symposium. Stanford University, California, USA, 2001.

[10] J. Broekstra, A. Kampman and F. van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. Proceedings of 1st ISWC, June 9-12th, 2002, Sardinia, Italy.

[11] F. Bry, T. Furche, P. Pâtrânjan and S. Schaffert. Data Retrieval and Evolution on the (Semantic) Web: A Deductive Approach. Proceedings of the Second International Workshop on Principles and Practice of Semantic Web Reasoning, St. Malo, France in September 2004.

[12] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson. Jena: Implementing the Semantic Web Recommendations. In 13th World Wide Web Conference, WWW2004, 2004.

[13] O. Corcho, A. Gómez-Pérez A., R. González-Cabero, and M. Suárez-Figueroa. ODEval: a Tool for Evaluating RDF(S), DAML+OIL, and OWL Concept Taxonomies. Proceedings of the 1st IFIP AIAI Conference. Toulouse, France.

[14] M. Cristani and R. Cuel. A Survey on Ontology Creation Methodologies. International Journal of Semantic Web and Information Systems (IJSWIS), Vol. 1, Issue 2.

[15] K. Dahlgren. Technical Overview of Cognition's Semantic NLP™ (as Applied to Search). Whitepaper.

[16] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R.V. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J.A. Tomlin, and J.Y. Zien. SemTag and Seeker: Bootstrapping the Semantic Web via automated Semantic Annotation. In *Proc. 12th International World Wide Web Conference*, Budapest, Hungary, pages 178-186, 2003

[17] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran. A Case for Automated Large Scale Semantic Annotation. *Journal of Web Semantics, 1*(1), 2003

[18] M. Fernández, A. Gómez-Pérez, J. Pazos, A. Pazos. Building a chemical ontology using MethOntology and the ontology design environment. IEEE Intelligent Systems Applications 1999; 4(1):37-45

[19] G. Friedrich G. and K. Shchekotykhin. A General Diagnosis Method for Ontologies. In Proceedings of the 4th International Semantic Web Conference (ISWC05), pages 232-246, 2005.

[20] A. Gómez-Pérez and M. Rojas-Amaya. Ontological Reengineering for Reuse. Proceedings of the 11th European Workshop on Knowledge Acquisition, Modeling and Management.

[21] T. Gruber. A Translation Approach to Portable Ontologies. In *Knowledge Acquisition*, Chapter 5(2), 1993

[22] N. Guarino and C. Welty. Evaluating Ontological Decisions with OntoClean. Communications of the ACM, 45(2) 2002, pp. 61-65

[23] Y. Guo, Z. Pan, and J. Heflin. LUBM: A Benchmark for OWL Knowledge Base Systems. Journal of Web Semantics 3(2), 2005, pp158-182.

[24] R.V. Guha, R. McCool. TAP: A Semantic Web Test-bed, *Journal of Web Semantics, 1*(1):81-87, 2003

[25] P. Haase P., F. van Harmelen, Z. Huang Z., H. Stuckenschmidt, and Y. Sure. A framework for handling inconsistency in changing ontologies. In Proceedings of ISWC2005, 2005.

[26] B. Hammond, A.P. Sheth, and K.J. Kochut. Semantic Enhancement Engine: A Modular Document Enhancement Platform for Semantic Applications over Heterogeneous Content. In *Real World Semantic Web Applications* (V. Kashyap and L. Shklar, eds.), Ios Press, pages 29-49, 2002

[27] S. Handschuh, S. Staab, and R. Studer. Leveraging Metadata Creation for the Semantic Web with CREAM. In *Proc. 26th Annual German Conference on AI*, Hamburg, Germany, pages19-33, 2003

[28] S. Handschuh, and S. Staab. CREAM CREAting Metadata for the Semantic Web. *Computer Networks, 42*:579-598, Elsevier, 2003

[29] E. Hatcher and O. Gospodnetic. Lucene in action. Manning Publications, 2005.

[30] W. Hersh and RT Bhupatiraju. "TREC Genomics Track Overview", In Proceedings of TREC 2003, pp. 14-23.

[31] M. Huang, X. Zhu, S. Ding, H. Yu, and M. Li. ONBIRES: ONtology-Based BIological Relation Extraction System. In the proceedings of the fourth Asian-Pacific Bioinformatics Conference. Taiwan, China 2006

[32] M. Janik, and K.J. Kochut. BRAHMS: A WorkBench RDF Store and High Performance Memory System for Semantic Association Discovery. In *Proc. Fourth International Semantic Web Conference*, Galway, Ireland, pages 431-445, 2005

[33] D. Klein and C. Manning. Accurate Unlexicalized Parsing. In: ACL. (2003) 423-430

[34] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of ACM, 46*(5):604-632, A.H.F. Laender, B.A. Ribeiro-Neto, A.S. da Silva, and J.S. Teixeira. A Brief Survey of Web Data Extraction Tools. *SIGMOD Record, 31*(2):84-93, 2002

[35] Y.L. Lee. Apps Make Semantic Web a Reality, *SD Times*, 2005

[36] M. Li, J. Badger, X. Chen, S. Kwong, P. Kearny, H. Zhang. An information-based sequence distance and its application to the whole mitochondrial genome phylogeny. Bioinformatics, 17:2 (2001).

[37] V. Lopez, M. Pasin, and E. Motta. AquaLog: An Ontology-portable Question Answering System for the Semantic Web. In the proceedings of the European Semantic Web Conference, Greece. 2005

[38] A. Lozano-Tello and A. Gomez-Perez. ONTOMETRIC: a method to choose the appropriate ontology. Journal of Database Management 2004.

[39] M. Maybury. New Directions in Question Answering. The MIT Press. November 2004.

[40] P. N. Mendes, B. McKnight, A. P. Sheth and J. C. Kissinger. "Enabling Complex Queries For Genome Data Exploration". The IEEE Second International Conference on Semantic Computing (ICSC) 2008 in Santa Clara California.

[41] G. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, *38*(11):39-41, 1995

[42] E. Miller. The Semantic Web is Here. In *Keynote at the Semantic Technology Conference*, San Francisco, California, USA, 2005

[43] H.-M. Muller, E. Kenny, and P. Sternberg. Textpresso: an ontology-based information retrieval and extraction system for biological literature. PLoS Biol., 2, 2003.

[44] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[45] B.Parsia, E. Sirin and A. Kalyanpur. Debugging OWL Ontologies. Proceedings of WWW 2005, May 10-14, 2005, Chiba, Japan.

[46] P. Paslaru et al. ONTOCOM: A Cost Estimation Model for Ontology Engineering. Proceedings of fifth ISWC, Athens, GA, USA. November, 2006.

[47] P. Plessers and O. De Troyer. Ontology Change Detection Using a Version Log. In Proceedings of the 4th ISWC, 2005.

[48] D. Rebholz-Schuhmann, H. Kirsch, M. Arregui M, et al. EBIMed-text crunching to gather facts for proteins from Medline. Bioinformatics (2007) 23:e237–44

[49] S.S. Sahoo, C. Thomas, A.P. Sheth, W.S. York, S. Tartir. Knowledge Modeling and its application in Life Sciences: A Tale of two Ontologies, In *Proc. 15th International World Wide Web Conference*, Edinburgh, Scotland, pages 317-326, 2006

[50] N.R. Shadbolt, T. Berners-Lee, and W. Hall. The Semantic Web Revisited. *IEEE Intelligent Systems, 21*(3):96–101, 2006

[51] A.P. Sheth, C. Bertram, D. Avant, B. Hammond, K.J. Kochut, and Y. Warke. Managing Semantic Content for the Web. *IEEE Internet Computing, 6*(4):80-87, 2002

[52] A.P. Sheth, and V. Kashyap. So Far (Schematically) yet So Near (Semantically). *IFIP Transactions a-Computer Science and Technology, 25*:283-312, 1993

[53] A.P. Sheth, and C. Ramakrishnan. Semantic (Web) Technology In Action: Ontology Driven Information Systems for Search, Integration and Analysis. *IEEE Data Engineering Bulletin, 26*(4):40-48, 2003

[54] A.P. Sheth. From Semantic Search & Integration to Analytics, In *Proc. Semantic Interoperability and Integration*, IBFI, Schloss Dagstuhl, Germany, 2004

[55] A.P. Sheth. Enterprise Applications of Semantic Web: The Sweet Spot of Risk and Compliance. In *Proc. IFIP International Conference on Industrial Applications of Semantic Web*, Jyväskylä, Finland, 2005

[56] H. Snoussi, L. Magnin, and J.-Y. Nie. Toward an Ontology-based Web Data Extraction. In Proc. Workshop on Business Agents and the Semantic Web, Calgary, Alberta, Canada, 2002

[57] K. Supekar, C.Patel and Y. Lee. Characterizing Quality of Knowledge on Semantic Web. Proceedings of AAAI FLAIRS, May 17-19, 2004, Miami Beach, Florida.

[58] S. Tartir, B. McKnight, and I. B. Arpinar. SemanticQA: Web-Based Ontology-Driven Question Answering. In the 24th Annual ACM Symposium on Applied Computing, Waikiki Beach, Honolulu, Hawaii, USA, March 8-12, 2009

[59] S. Tartir, and I.B. Arpinar. Ontology Evaluation and Ranking using OntoQA. In Proc. First IEEE International Conference on Semantic Computing, Irvine, California, USA, 2007

[60] S. Tartir, I.B. Arpinar, M. Moore, A.P. Sheth, and B. Aleman-Meza. OntoQA: Metric-Based Ontology Quality Analysis. In *Proc. IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*, Houston, TX, USA, 2005

[61] S. Tartir, I. B. Arpinar and A. P. Sheth. Ontological Evaluation and Validation. In R. Poli (Editor): Theory and Applications of Ontology (TAO), volume II: Ontology: The Information-science Stance Springer, 2008

[62] M. Vargas-Vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt, and F. Ciravegna. MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup. In *Proc. 13th International Conference on Knowledge Engineering and Management*, Sigüenza, Spain, 2002

[63] C. Wang, M. Xiong, Q. Zhou, Y. Yu. Panto - a portable natural language interface to ontologies. In: 4th ESWC, Innsbruck, A, pp. 473–487 (2007)

[64] L. Xiao, L. Zhang, G. Huang, B. Shi. Automatic Mapping from XML Documents to Ontologies, In *Proc. 4th International Conference on Computer and Information Technology*, Wuhan, China, 2004