OntoKhoj: A Semantic Web Portal for Ontology Searching, Ranking and Classification

Chintan Patel, Kaustubh Supekar, Yugyung Lee, E.K. Park
School of Computing and Engineering
University of Missouri-Kansas City
{copdk4, kss2r6, leeyu, ekpark@umkc.edu}

ABSTRACT

The goal of the next generation Web is to build virtual communities, wherein software agents and people can work in cooperation by sharing knowledge. To achieve this goal, the emerging Semantic Web community has proposed ontologies to express knowledge in a machine understandable way. The process of building and maintaining ontologies, which is known as Ontology Engineering, presents unique challenges. These challenges are related to lack of trustworthy and authoritative knowledge sources and absence of centralized repository to locate ontologies to be reused. In this paper, we propose a Semantic Web portal, called OntoKhoj that is designed to simplify the Ontology Engineering process. The methodology in developing OntoKhoj is based on algorithms used for searching, aggregating, ranking and classifying ontologies in Semantic Web. The proposed OntoKhoj would 1) allow agents and ontology engineers to retrieve trustworthy, authoritative knowledge, and 2) expedite the process of ontology engineering through extensive reuse of ontologies. We have implemented the OntoKhoj portal and further validated our system on the real ontological data in the Semantic Web.

1. INTRODUCTION

The Semantic Web is an emerging field, with the aim of building infrastructure, wherein software agents and people can work in cooperation by sharing knowledge [1]. This requires incorporating machine understandable information into the Web designed solely for human consumption. With the support of a new set of solutions developed by the Semantic Web community to meet this requirements, more Web content represented in ontologies would be accessible to machines. The process of building and maintaining ontologies in the Semantic Web, which is known as Ontology Engineering, presents unique challenges.

In this paper, we try to tackle the two major challenges: 1) Searching ontologies and 2) Trusting Information over the Semantic Web. First, the Semantic Web is facing the same problems encountered by WWW during its nascent stages, namely searching relevant information over Web. In the Semantic Web, ontologies represent the knowledge to be shared by formally defining concepts and relations of entities occurring in domain or universe of discourse. Some of the most common questions raised by the Semantic Web community include *Where can I start with creating an application over the Semantic Web?*, *Where can I find ontologies for my domain of interest?* and *How to build ontologies using such publicly available ontologies?* Interestingly, they raise some of the most significant problems inherent in distributed computing environment.

Technically speaking, the growth and success descriptor of Semantic Web could be the number of ontologies present on the Web. Although there are an increasing number of ontologies, its proliferation is quite slow, compared to the Web pages in the traditional Web. We believe that the major obstacles exist in the process of conceptual modeling and Ontology Engineering, an arduous and exceedingly intricate task that requires specialized design skills as well as comprehensive domain knowledge. It is noteworthy that Semantic Web does allow ontologies to be distributed, reusable and extendable; different knowledge sources, sketching the same domain, can be present anywhere over the Web and existing knowledge can be reused or extended for different domains. Specifically, cross-referenced and hyperlinked ontologies make it easy to model a Web of concepts and relationships. The first step toward such an improvement is building a portal that can search relevant ontologies over the Semantic Web.

Another issue is *How much trust we are placing in the information present on the Semantic Web.* In an independent environment such as Web, where there are no restrictions on the information being published, it becomes the liability on the part of the consumer to accurately judge the quality and validity of the information provider. The information is provided by many different sources and more importantly, the information in Semantic Web has been envisioned for not only human beings but machine. This implies machines are responsible, to a certain degree, for discerning the trust of information source. Currently, there are no appropriate solutions to characterize validity and quality of ontologies.

We are highly motivated by the fact that having a dynamic and trustworthy ontology information source is extremely important in advancement and growth of Semantic Web. To visualize the solutions to aforementioned problems we draw an analogy from current Web, ontologies in Semantic Web are akin to Web pages connected to each other using hyperlinks aka relationships (rdf:about, rdfs:subClass). In current Web, information searching and indexing is performed by specialized search engines (e.g., Google.com, Altavista.com) using proprietary algorithms to crawl and rank the Web pages and subsequently allowing users to perform simple query, keyword based searches. In similar lines we have developed a Semantic Web portal, OntoKhoj, that would provide services related to searching, ranking, aggregating and classifying ontologies crawled from the Semantic Web, thereby providing Knowledge Engineers and software agents, a source for authoritative, trustworthy ontologies.

2. RELATED WORK

Growth of Semantic Web has led to massive growth in the use and development of ontology. The central idea of Ontology Engineering in Semantic Web is extensive reuse of existing ontologies. Currently, Semantic Web doesn't have any infrastructure that allows Knowledge Engineers to search and peruse relevant domain ontologies. Lack of central index of ontologies aggravates the problem. Recently, several tools for Ontology Engineering have been developed; Protégé-2000 [19], OntoEdit [15], OilEd [14]. Other related tools have been built for ontology merging-PROMPT [13], ontology access OKBC [3], and KAON [10]. Moreover, these tools do not provide any facilities for Knowledge Engineers to share or collaborate and reuse their work.

Ranking based on citations has been a major area of research [11, 5]. Googles [6] PageRank is one of finest example that shows the success of the citation algorithms [17] in Web environment. OntoKhoj extends the functionality of PageRank to handle the critical issues that arise while considering Web of ontologies rather than simple Web of HTML (Hypertext Markup Language) pages.

Siebes et al. [22] describe a mechanism wherein each Agent ranks and stores statements received from its peer to determine the validity/trust of the information source, but the major problem in such approach is related to scalability. It becomes practically impossible for a single Agent to store and keep track of all information of its past transactions.

The closest similar work is Web-KB2 [24], that allows users to retrieve, re-use, complement, annotate and be guided by other users' knowledge. However, it is based on propriety Knowledge representation schemes and non-conformant to W3C standardized RDF based representation model.

Another significant initiative in this area is Ontolingua by KSL Stanford [16] that gives a distributed collaborative environment to browse, create, edit, modify, and use ontologies, but it requires the user to register and then publish ontologies. OntoKhoj uses Web and Semantic Web crawling techniques to retrieve ontologies thereby preserving spirit of openness and independence of publishing information over Web.

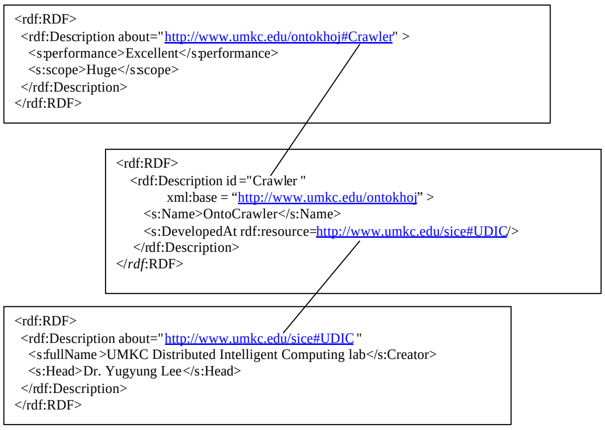
3. THE ONTOKHOJ MODEL

3.1 Ontology Crawling

To perform *knowledge crawling*, which would be quite different from Web page crawling, requires consideration of specific features in the underlying knowledge representation framework. Semantic Web essentially exploits the concept of URI [8] to represent the knowledge in form of a simple triple VSO = {Value, Subject, Object}, which accounts for its scalability and portability over Internet . The data model defined for this purpose is RDF (Resource Description Format)[20], wherein each RDF entity has an associated URI, that allows citing and reusing, thereby accelerating the proliferation of knowledge. Recently, various ontology languages (RDFS, DAML+OIL, OWL) that are based on RDF data model have been standardized.

One of the major issue to be considered while crawling RDF data is the nature of *hyperlinking* existing in the RDF model. Basically, RDF uses the generic and abstraction notion of URI to describe RDF entities that doesn't guarantee any physical presence of the resource. This implies that one can give any hypothetical URI, for example http://www.uspresidents.com/#ME, which may not be necessarily present on Web. Moreover, the RDF data can be present in many different forms (e.g. as physical RDF document at a given URL or as RDF annotation embedded in HTML pages).

The following RDF example (Figure 1) shows the referencing methodology that is inherent in the RDF data model.



```
<rdf:RDF>
 <rdf:Description about="http://www.umkc.edu/ontokhoj#Crawler" >
   <s:performance>Excellent</s:performance>
   <s:scope>Huge</s:scope>
 </rdf:Description>
</rdf:RDF>
```

```
<rdf:RDF>
   <rdf:Description id ="Crawler "
           xml:base = “http://www.umkc.edu/ontokhoj” >
      <s:Name>OntoCrawler</s:Name>
      <s:DevelopedAt rdf:resource=http://www.umkc.edu/sice#UDIC/>
    </rdf:Description>
</rdf:RDF>
```

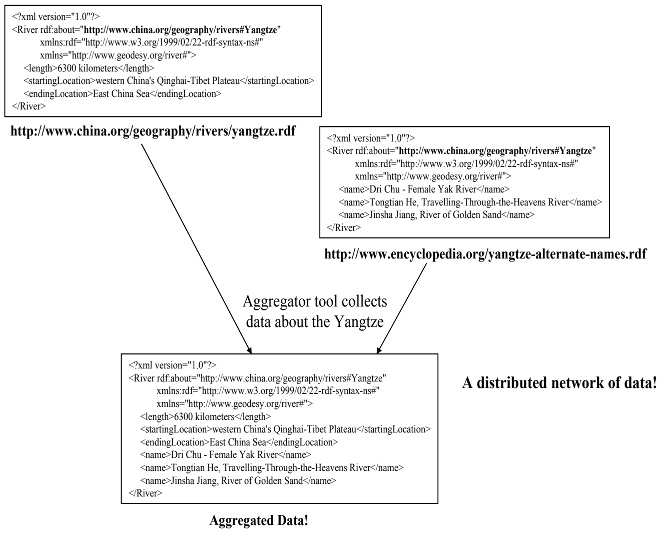
```
<rdf:RDF>
 <rdf:Description about="http://www.umkc.edu/sice#UDIC "
   <s:fullName >UMKC Distributed Intelligent Computing lab</s:Creator>
   <s:Head>Dr. Yugyung Lee</s:Head>
 </rdf:Description>
</rdf:RDF>
```

Figure 1: The Ontology Crawling: Hyperlinked RDF Fragments

The Semantic Web allows ontologies to be distributed i.e. there can be several RDF chunks belonging to same logical URI but physically present at different location. After crawling, we aggregate those chunks into a single ontology depending upon the URI of the concepts. For example, in figure 2, two fragments located at different URLs belonging same namespace http://www.china.org/geography/rivers#Yangtze are eventually aggregated.

Here are the primary features that our ontology crawling method provides:

- Perform ontology crawling on heterogenous Web sources including HTML, XML, RDF, DAML+OIL, OWL



```
<?xml version="1.0"?>
<River rdf:about="http://www.china.org/geography/rivers#Yangtze"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://www.geodesy.org/river#">
   <length>6300 kilometers</length>
   <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
   <endingLocation>East China Sea</endingLocation>
</River>
```

```
<?xml version="1.0"?>
<River rdf:about="http://www.china.org/geography/rivers#Yangtze"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://www.geodesy.org/river#">
   <name>Dri Chu - Female Yak River</name>
   <name>Tongtian He, Travelling-Through-the-Heavens River</name>
   <name>Jinsha Jiang, River of Golden Sand</name>
</River>
```

```
<?xml version="1.0"?>
<River rdf:about="http://www.china.org/geography/rivers#Yangtze"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://www.geodesy.org/river#">
   <length>6300 kilometers</length>
   <startingLocation>western China's Qinghai-Tibet Plateau</startingLocation>
   <endingLocation>East China Sea</endingLocation>
   <name>Dri Chu - Female Yak River</name>
   <name>Tongtian He, Travelling-Through-the-Heavens River</name>
   <name>Jinsha Jiang, River of Golden Sand</name>
</River>
```

http://www.china.org/geography/rivers/yangtze.rdf
http://www.encyclopedia.org/yangtze-alternate-names.rdf
Aggregator tool collects data about the Yangtze
A distributed network of data!
Aggregated Data!

Figure 2: The Ontology Aggregation (an example from www.xfront.com)

- Avoid circular links (of RDF URIs)
- Distributed crawling (running parallel threads with different seeds)
- Aggregating RDFs chunks (belonging to same URI) (See Figure 2)

3.2 Ontology Classification

We perform ontology Classification to *fit* the ontologies into a predefined directory of general categories. Traditional classification techniques have been applied to domain of Web Page classification [18], gene classification. They rely on machine learning algorithms such as Naïve Bayes, K Nearest Neighbor.

We believe it is quite a novel and challenging task to develop a new data mining paradigm for ontology classification (Note it is not ontology based classification but *ontology classification* itself). Such a paradigm would take into consideration various semantic relationships and spatial distribution of concepts in the ontology. Further discussion is out of scope for this paper and is part of our future work.

In our work, we made couple of observations related to ontologies:

- Ontologies are generally a representative of a particular domain. They render both highly specific and generic information explicitly relevant to the universe of discourse.
- Ontologies capture most of the common terminologies for the given domain.

Based on aforementioned observations we can consider terms describing concepts and relationships in ontologies as plain text, so the problem boils down to that of text classification. Hence, we can apply several traditional classification algorithms and tools [2]. In OntoKhoj, the classifier was trained by the initial training data derived from plain categorized source [4] containing huge number of manually classified datasets. Each crawled and aggregated ontology is handed over to the classifier and it determines whether a new ontology belongs to a particular topic with sufficient confidence. The classified ontologies are stored into the corresponding directory which can be graphically explored for user's queries, and also can be traversed or retrieved by agents. We will show the experimental results confirm the effectiveness of the proposed classification approach in Section 5.

Priority (Weight)	Relationship	Language Specific
1	instantiation	rdf:type
2	subClass	rdfs:subclass, daml:subClass
3	domain/range	rdfs:domain, daml:range

Table 1: Weights of Hyperlinks

3.3 Ontology Ranking

In huge pool of information available over Web, Ranking plays a major role in providing a authoritative and useful information to third party users. With Internet being a distributed and open environment, some malicious person can manipulate the weaknesses in the ranking algorithm to boost his/her rank. In fact, currently there are many companies specializing in business of exploiting the ranking algorithms to manipulate the ranking in top Search engines.

To have fair and just rankings, the ranking algorithm should be designed in a tamper proof manner. The algorithm should bring out the most authoritative information source in higher rank. Same reasoning holds true for the knowledge in Semantic Web. The basic idea is to associate a rank to all the concepts in the ontology being cached. We believe that OntoRank algorithm would automatically bubble up the trustworthy ontologies on Semantic Web. The reason is analogous to "citation" in the scientific publishing, the more a paper is cited by others indicates higher quality and trustworthiness of the content in the paper. Moreover, this trust is also being propagated creating a chain of trust.

Considering the richness of ontology modeling languages, the Semantic Web is much more complex than traditional hyperlinked Web pages. Hence we focused our attention on the type of linking (relationships) which can exist given the current set Semantic Web languages (RDF, RDFS, DAML+ OIL, OWL). The various types of hyperlinking across ontologies have some inherent semantics that could be exploited to determine the importance of a given link in RDF Web graph. For example, if a knowledge engineer subclasses an ontology concept (developed by someone else), it indicates that he/she uses some original features but still wants to add some new features. Similarly, if one instantiates an ontology concept directly, it indicates a complete endorsement. Referring to a concept in other ontology as domain/range would assume least priority. Hence, as depicted in Table 1, we prioritized such semantic relationships based on the intuitive reasoning.

Crawler fetches the RDF documents according to the physical links (HTML URLs, RDF URIs). However, since URI may not necessarily point to actual physical Resource, we modeled an overlay logical layer consisting of hyperlinked RDF ontologies. Based on a logical layer of resources, we propose our ontology ranking methodology which considers a concept of *Referencing*. For a given concept, C_i in ontology O_i, and the ontology referencing hyperlinks $Ref \in \{$rdf:type, rdfs:subclass, daml:subClass, rdfs:domain, rdfs:range, rdf:seeAlso, rdf:about$\}$. We define the following terms:

- The identity and rank of the referrer, $Ref(?C,\ C_i)$ where $?C$ is a undetermined concept.
- The number of citations by others, $|Ref\ (?C,\ C_i)|$
- The distance of reference from the origin to the target, $Dist(Ref\ (C_o,\ C_d))$ where a chain of referring exists from C_o to C_d (e.g., $Ref\ (C_o,\ C_i)$, $\ldots$, $Ref(C_j,\ C_d)$) and avoids circular reference $Ref\ (C_o,\ C_d)$ and $Ref\ (C_d,\ C_o)$. Thus, the distance of a direct referencing is equal to 1. We place less weight on concepts that are further apart through the reference links.

Our work is influenced by the PageRank algorithm [17] which measures its citation importance using maps containing minimum 518 million hyperlinks, and prioritizes the results of keyword-based searches in the Google system [6]. We have developed an algorithm OntoRank that assigns a rank to an ontology in Semantic Web. Our work is different from the PageRank [17] in several aspects such as considering different types of link and additional constraints like distances. Hence apart from considering the rank of the referrer, we also take into consideration the weight of the type of reference (relationship).

We give a formal treatment to the aforementioned methodology. Let O be the ontology whose rank we wish to determine. Let α be the number of ontologies referring O, each of the referring ontologies can have more than one referrals to ontology O. Let β_i be the number of referrals from ontology O_i to O. Let Ω_l be the total number of outgoing referrals from ontology O_i. Let T be the weight of the reference, N be the normalization factor. The OntoRank, $OR(O)$ is defined as follows:

$$OR(O) = N * \sum_{l=1}^{\alpha}(1/\Omega_l) * \sum_{j=1}^{\beta_i} OR(O_i) * T_j \qquad (1)$$

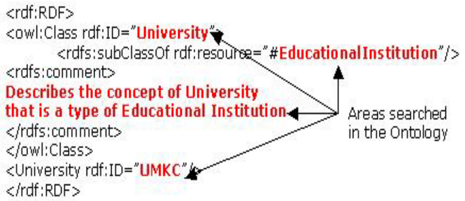
We believe that the simplicity of the proposed algorithm accounts for its scalability and stability. We forego many other issues related to circular references, dangling links for the sake of brevity.

4. THE ONTOKHOJ SYSTEM

4.1 Ontology Search

With the proliferation of ontologies over the Web, one of the challenging tasks is to search for the desired ontologies. To the best of our knowledge, there are no engines that can search ontologies on behalf of the users of Semantic Web (i.e., humans and machines). OntoKhoj is a major step towards realizing the promise that a search engine for distributed ontologies provides. The OntoKhoj search engine extends the traditional approach (keyword-based search) to cover the information in Semantic Web. More specifically, the OntoKhoj engine retrieves information from following areas in the serialized RDF model (depicted in Figure 3):

- Concept Name
- Content in special tags such as <rdfs:comment> (if present)
- All the Literals pointed by a particular Subject.



```
<rdf:RDF>
<owl:Class rdf:ID="University">
        <rdfs:subClassOf rdf:resource="#EducationalInstitution"/>
<rdfs:comment>
Describes the concept of University
that is a type of Educational Institution
</rdfs:comment>
</owl:Class>
<University rdf:ID="UMKC"/>
</rdf:RDF>
```

Figure 3: Search Areas in Ontology

Studies show that users prefer simple keyword-based interfaces over complex query based search interface. However, a simple keyword-based query does not give enough information to determine the right context for the query, leading to poor precision in results. In order to meet the requirements, OntoKhoj provides several different interfaces to satisfy search requirements of different users at different conceptualization level.

Context based Query Interface: Our approach for constructing the context based query interface is based on three dictionary entries: senses, synonym, and hyponyms. This approach is based on WordNet, which is a lexical ontology developed by Miller et al. [12]. The OntoKhoj portal has an interface with WordNet lexical reference system, which is responsible for the retrieval and display of the dictionary entities. The operation of Context based Query Interface in the OntoKhoj portal is summarized in following steps.

- The search interface allows the user to disambiguate senses by selecting a sense from the displayed listing of various senses associated with the keyword. A concept can have different meaning in different context (e.g., concept date could be referred to as *day of the month* or date - *sweet edible fruit of the date palm with a single long woody seed*).
- For the selected sense of the keyword, associated synonym and hypernym (taxonomy) terms are retrieved from the WordNet.
- If the search term is not an exact match for any of the concepts in the ontologies, the closest (synonym) matching is performed.

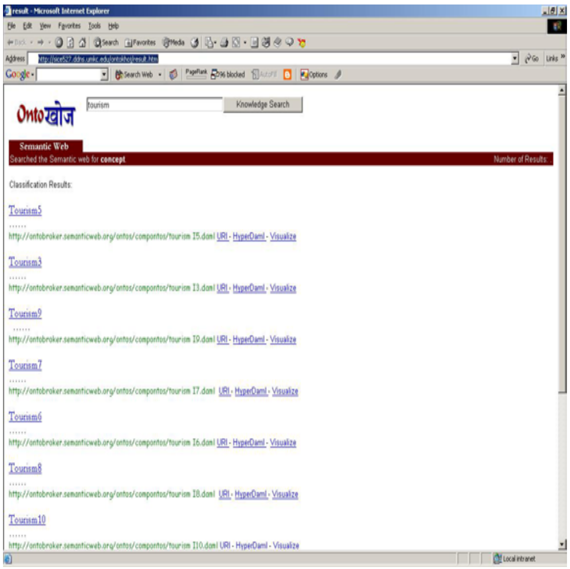
- If the search term is not found, then the hypernymic matching is performed. It traverses the hypernymic link upward until a term, which is close to the keyword of interest, is found.

OntoKhoj Machine Interface: Semantic Web is meant for agents to interpret information on Web in lieu for humans, in this spirit we need to automate the process of searching ontologies and possibly interpret information on users behalf.

- An interface for agents to access and query the directory of classified ontologies is provided. The directory is represented in RDF ontology that allows agents to automatically traverse and retrieve desired information.
- Advanced Logic query interfaces (e.g. RDQL, FLogic) allow to specify search constraints thereby providing sophisticated inferencing capabilities across ontologies.

4.2 Implementation

We have implemented OntoKhoj, a Semantic Web Portal, that is designed to simplify the Ontology Engineering process. The implementation methodology is based on algorithms, used for searching, crawling, classifying and ranking ontologies in Semantic Web. In current Semantic Web, multiple ontologies describing a same domain/concept appear to be quite common. Responding to the urgent needs of the Semantic Web in the current context, the implemented OntoKhoj portal 1) allows agents and ontology engineers to retrieve trustworthy, authoritative knowledge, and 2) expedites the process of Ontology Engineering through extensive reuse of ontologies. The tool is currently accessible through our website at http://sice527.ddns.umkc.edu/ontokhoj.


result - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address
Go Links
Google Search Web PageRank 596 blocked AutoFill Options
Knowledge Search
Semantic Web
Local intranet

http://sice527.ddns.umkc.edu/ontokhoj/result.htm

tourism

Searched the Semantic web for **concept**

Number of Results:

Classification Results:

Tourism5
......
http://ontobroker.semanticweb.org/ontos/compontos/tourism I5.daml URI - HyperDaml - Visualize

Tourism3
......
http://ontobroker.semanticweb.org/ontos/compontos/tourism I3.daml URI - HyperDaml - Visualize

Tourism9
......
http://ontobroker.semanticweb.org/ontos/compontos/tourism I9.daml URI - HyperDaml - Visualize

Tourism7
......
http://ontobroker.semanticweb.org/ontos/compontos/tourism I7.daml URI - HyperDaml - Visualize

Tourism6
......
http://ontobroker.semanticweb.org/ontos/compontos/tourism I6.daml URI - HyperDaml - Visualize

Tourism8
......
http://ontobroker.semanticweb.org/ontos/compontos/tourism I8.daml URI - HyperDaml - Visualize

Tourism10
......
http://ontobroker.semanticweb.org/ontos/compontos/tourism I10.daml URI - HyperDaml - Visualize

Figure 4: OntoKhoj: Ontology Search Result Page

The prototype system of the OntoKhoj portal was implemented using Java on Linux platform. The four major functionalities include 1) Crawling ontologies over the Web, 2) Indexing and Ranking these crawled ontologies in local repository, 3) Classifying each of the stored ontology, 4) Ontology visualization.

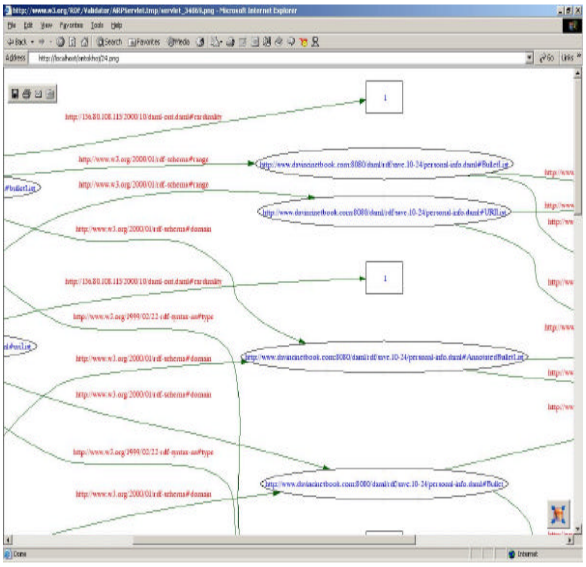

http://www.w3.org/RDF/Validator/ARPServlet.tmp/servlet_34869.png - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/ontokhoj/24.png
Go Links
1
http://136.80.108.115/2000/10/daml-ont.daml#cardinality
http://www.w3.org/2000/01/rdf-schema#range
http://www.davincinetbook.com:8080/daml/rdf/rdfsave.10-24/personal-info.daml#BulletList
#bulletList
http://www.w3.org/2000/01/rdf-schema#range
http://www.davincinetbook.com:8080/daml/rdf/rdfsave.10-24/personal-info.daml#URIList
http://www.w3.org/2000/01/rdf-schema#domain
1
http://136.80.108.115/2000/10/daml-ont.daml#cardinality
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
#uriList
http://www.w3.org/2000/01/rdf-schema#domain
http://www.davincinetbook.com:8080/daml/rdf/rdfsave.10-24/personal-info.daml#AnnotatedBulletList
http://www.w3.org/2000/01/rdf-schema#domain
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
http://www.davincinetbook.com:8080/daml/rdf/rdfsave.10-24/personal-info.daml#Bullet
http://www.w3.org/2000/01/rdf-schema#domain
Done
Internet

Figure 5: OntoKhoj: Ontology Visualization Page

The first task of crawling ontologies was accomplished through RDF crawler, which combines advanced features of [21] and [9]. Our crawler can retrieve ontologies represented in RDF, RDF embedded HTML and DAML+OIL format. The crawled ontologies are stored in a local repository, which is a MySQL database. As the second task, we indexed all the crawled ontologies, subsequent implementation of the OntoRank helped in determining authoritative ontologies. The OntoRank algorithm is described in detail in Section 3.3 and the ranked ontologies for the Tourism domain are shown in Figure 4.

The ontology classification of the OntoKhoj portal has been implemented using Rainbow [2], a document classification tool. The tool supports implementation of four classification algorithms - Naïve Bayes, TFIDF/Rocchio, Probabilistic Indexing and K-Nearest neighbor. To obtain training data set, we employed an intuitive approach. DMOZ [4], an open directory project, provides a classification of Web pages into 460,000 categories. Every category listed in DMOZ directory has associated collection of Web pages (about 100 pages). Our Java based implementation extracts text from the web pages of the DMOZ category and trains the Rainbow tool by feeding all the extracted pages from the 460,000 classes/categories. Each of the ontology stored in the local repository is manually entered into the trained Rainbow tool; subsequent testing yields a classification of the selected ontology. Finally, our visualization tool, implemented based on GraphViz [7] converts the classified ontologies into a visual representation (as shown in Figure 5) .

5. EXPERIMENTAL RESULTS AND EVALUATION

5.1 Ontology Crawling and Aggregation

In this section we describe the methodology for obtaining ontological data, which form the basis of our experimental analysis. The extended version of the RDF crawler, as described in Section 3.1, was implemented in Java on the Linux platform. The execution of RDF crawler for 48 hours yielded considerable amount of data, detailed statistics provided in Table 2. Limited computational resources restricted the execution period of RDF crawler to 48 hours. Further, as the number of publicly available ontologies is limited, we consider the dataset of 418 ontologies as a good representative of the entire population.

Number of Web pages visited	2018412
Number of Concepts crawled	19870
Number of Relationships Discovered	1321
Total Ontologies (after Aggregation)	418

Table 2: OntoKhoj statistics

5.2 Ontology Classification

We have performed a series of experiments to determine the most suitable algorithms for the ontology classification. For this purpose, we selected four popular classification algorithms - Naïve Bayes, TFIDF, KNN and PRIND. For the testing dataset, 22 ontologies were selected from five overlapping domain of interests: Sports, Baseball, Soccer, University, and Computer Science. The subject of our interest is the selection criterion, ontologies with a certain degree of overlapping domain were chosen. Each of the 22 ontologies was manually entered into the trained Rainbow Tool [2] to generate classification accuracy for each of the four classification algorithms.

Because our dataset of twenty two ontologies is small, we computed a desired classification result of the 22 ontologies by hand. Then we used *Sensitivity* and *Specificity* to compare the results. For evaluation purposes, we use the following common terms. True Positives (TP) indicates the classification algorithm classifies an input ontology as a domain X, and this classification is considered valid according to a human expert (*Correctly classified*). True Negatives (TN) indicates the classification algorithm does not classify an ontology as a domain X, and the classification is agreed by a human expert(*Correctly unclassified*). False Positives (FP) indicates the classification algorithm classifies an ontology as a domain X, but a human expert would not consider the classification valid (*Incorrectly classified*). False Negatives (FN) indicates the classification algorithm does not classify an input ontology as a domain X, but it should be classified, according to a human expert (*Missed classification*)

We can now use the Equation 2 to compute the Sensitivity (Sn) and the Equation 3 to compute the Specificity (Sp) of our experiment. Sn is described as the proportion of true positives a test detects of all the positives. Sp is described as the proportion of true negatives a test detects of all the negatives.

$$Sn = TP/(TP + FN) \tag{2}$$

$$Sp = TN/(TN + FP) \tag{3}$$

Results were generated from the classification algorithms on the selected 22 ontologies. Since ontologies are representative of a domain, it is supposed to be uniquely classified by the algorithms. Particulary, it is interesting to see overlapping ontologies were correctly classified to a certain degree. Considering an example, *CS Department* ontology must be classified as *Computer Science* domain rather than *University* domain. Table 3 shows the relevant statistics obtained.

It is shown via experiments that all the algorithms perform well in classifying ontologies. However, it is interesting to see how the algorithms work for highly overlapping domain ontologies. Lack of Ontological data constrained us from performing such critical tests. However, based on the experimental analysis, Naïve Bayes classification algorithm performed relatively well. Hence, Naïve Bayes was found to be the most suitable algorithm for the OntoKhoj classification implementation.

5.3 Ontology Ranking

For the given classified ontologies, the OntoRank algorithm (as described in Section 3.3) subsequently ranks them in descending order of their rank. For performing experiments with the proposed ranking algorithm, we obtained 10 ontologies in Tourism domain through OntoKhoj search interface. A subsequent execution of the algorithm on the dataset yielded results - a ranking of 10 tourism ontologies (as show in Figure 4). A subjective evaluation of the results confirmed the correctness of the OntoRank algorithm. We admit that the subjective interpretation of our results is limited. Our research in the similar direction [23] focused on the development of metric-based ontology ranking method considering the preferences of users. In future, we would like to incorporate a dynamic approach, wherein agents or users would express their own preference through the OntoKhoj portal for raking ontologies. We foresee that user oriented mechanism for ontology ranking would help in improving the practical accuracy of the results.

6. CONCLUSION

Responding to the compelling requirements of the Semantic Web community, we developed the OntoKhoj portal, which assists humans by simplifying the process of Ontology Engineering. The OntoKhoj development is based on novel methodologies allowing advanced searching, ranking, aggregating and classifying of ontologies crawled from the Semantic Web. We focused on developing a proof-of-concept prototype of the proposed models and testing it on real Semantic Web data. Our claims are supported by experimental results of ontology crawling, ranking, classification, carried out with ontology data obtained from the Semantic Web. We believe that our OntoKhoj Web portal will provide knowledge engineers and agents, a source for authoritative and trustworthy ontologies on Semantic Web and expedite the process of Ontology Engineering through extensive reuse of ontologies.

7. REFERENCES

[1] Berners-Lee, T., Hendler, J., Lassila, O. The Semantic Web: A new form of Web content that is meaningful to

Domain	Classification Algorithm	TP	TN	FP	FN	Sensitivity	Specificity
University	Naïve Bayes	6	16	0	0	1.0	1.0
	TFIDF	4	15	1	2	0.66	0.9375
	KNN	5	15	1	1	0.833	0.9375
	PRIND	4	16	2	0	1.0	0.889
Computer Science	Naïve Bayes	4	18	0	0	1.0	1.0
	TFIDF	3	16	2	1	0.75	0.89
	KNN	3	17	1	1	0.75	0.944
	PRIND	2	18	0	2	0.5	1.0
Sports	Naïve Bayes	5	15	2	0	1.0	0.882
	TFIDF	5	13	3	1	0.833	0.8125
	KNN	5	17	0	0	1.0	1.0
	PRIND	3	17	2	0	1.0	0.89
Baseball	Naïve Bayes	2	19	0	1	0.66	1.0
	TFIDF	1	19	0	2	0.33	1.0
	KNN	2	19	0	1	0.66	1.0
	PRIND	1	19	1	1	0.5	0.95
Soccer	Naïve Bayes	3	18	0	1	0.75	1.0
	TFIDF	2	18	0	2	0.5	1.0
	KNN	4	17	1	0	1.0	0.94
	PRIND	3	17	1	1	0.75	0.94

Table 3: Ontology Classification Statistics

computers will unleash a revolution of new possibilities. In Scientific American, Mai 2001. Online at: http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html

[2] Bow: Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering, http://www-2.cs.cmu.edu/ mccallum/bow/

[3] V. K. Chaudhri, A. Farquhar, R. Fikes, P. D. Karp, and J. P. Rice: Open Knowledge Base Connectivity 2.0, Knowledge Systems Laboratory, KSL-98-06, January 1998. http://wwwksl-svc.stanford.edu:5915/doc/project-papers.html

[4] DMOZ: Open Directory Project Home Page, http://www.dmoz.org/

[5] E. Garfield, Citation Indexing: Its Theory and Application in Science, Technology, and Humanities, John Wiley & Sons, New York, 1979.

[6] Google Search Engine Home Page, http://www.google.com/

[7] GraphViz, Graph Drawing Software, http://www.research.att.com/sw/tools/graphviz/

[8] IETF RFC 2396 Uniform Resource Identifiers (URI): Generic Syntax. http://www.ietf.org/rfc/rfc2396.txt

[9] HP Jena Semantic Web Toolkit, http://www.hpl.hp.com/semweb/jena.htm

[10] KAON Home Page. http://kaon.semanticweb.org/

[11] S. Lawrence, C. L. Giles, K. Bollacker, Digital Libraries and Autonomous Citation Indexing, IEEE Computer, Volume 32, Number 6, pp. 67-71, 1999. http://www.neci.nec.com/ lawrence/papers/aci-computer98/aci-computer99.html

[12] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. Miller; ”Introduction to WordNet: An On-Line Lexical Database”; http://www.cosgi.princeton.edu/ wn; August 1993.

[13] Noy, N. F. and M. A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. in Seventeenth National Conference on Artificial Intelligence (AAAI2000).

[14] OilEd Website, http://oiled.man.ac.uk/

[15] OntoEdit Website, http://www.ontoknowledge.org/tools/ontoedit.shtml

[16] Ontolingua Website, http://www.ksl.stanford.edu/software/ontolingua/

[17] L. Page, S. Brin, R. Motwani, and T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web, January 1998.

[18] R. Prabowo, M. Jackson, P. Burden, H. Knoell, Ontology-Based Automatic Classification for the Web Pages: Design, Implementation and Evaluation, In Proceedings of the Third International Conference on Web Information Systems Engineering (WISE'00) December 12 - 14, 2002 Singapore

[19] Protégé Project Homepage, http://protege.stanford.edu/

[20] Resource Description Framework (RDF) Home Page, http://www.w3.org/RDF/

[21] RDF Crawler, http://ontobroker.semanticweb.org/rdfcrawl/help/specification.ht

[22] R. Siebes and F. van Harmelen, Ranking agent statements for building evolving ontologies. In

Proceedings of the AAAI-02 workshop on meaning negotiation, Alberta, Canada, July 28 2002

[23] K. Supekar, C. Patel, Y. Lee, E.K. Park, Characterizing Quality of Knowledge on Semantic Web, University of Missouri - Kansas City, Technical Report, 2003.

[24] The Web-KB Website, http://meganesia.int.gu.edu.au/ phmartin/WebKB/