Finding and Ranking Knowledge on the Semantic Web *

Li Ding, Rong Pan, Tim Finin, Anupam Joshi, Yun Peng, and Pranam Kolari

Department of Computer Science and Electrical Engineering,
University of Maryland, Baltimore County, Baltimore MD 21250
{dingli1,panrong1,finin,joshi,ypeng,kolari1}@cs.umbc.edu

Abstract. Swoogle helps software agents and knowledge engineers find Semantic Web knowledge encoded in RDF and OWL documents on the Web. Navigating such a Semantic Web on the Web is difficult due to the paucity of explicit *hyperlinks* beyond the namespaces in URIrefs and the few inter-document links like rdfs:seeAlso and owl:imports. In order to solve this issue, this paper proposes a novel Semantic Web navigation model providing additional navigation paths through Swoogle's search services such as the *Ontology Dictionary*. Using this model, we have developed algorithms for ranking the importance of Semantic Web objects at three levels of granularity: documents, terms and RDF graphs. Experiments show that Swoogle outperforms conventional web search engine and other ontology libraries in finding more ontologies, ranking their importance, and thus promoting the use and emergence of consensus ontologies.

1 Introduction

As the scale and the impact of the World Wide Web has grown, search engines have assumed a central role in the Web's infrastructure. Similarly, the growth of the Semantic Web will also generate a need for specialized search engines that help agents[1] find knowledge encoded in Semantic Web languages such as RDF(S) and OWL. This paper discusses two important aspects of Semantic Web search engines: helping agents *navigate*[2] the Semantic Web and ranking search results.

The utility of Semantic Web technologies for sharing knowledge among agents has been widely recognized in many domain applications. However, the Semantic Web itself (i.e., the unified RDF graph comprised of many decentralized online knowledge sources) remains less studied. This paper focuses on the Semantic Web materialized as a collection of **Semantic Web Documents** (SWDs)[3] because web pages are well known as the building blocks of the Web.

* Partial support for this research was provided by DARPA contract F30602-00-0591 and by NSF awards NSF-ITR-IIS-0326460 and NSF-ITR-IDM-0219649.

[1] The term *agents* refers to programs, tools, and human knowledge engineers that might use Semantic Web knoweledge.

[2] The term *navigation* refers to a process of following a series of links (explicit or implicit) from an initial starting point to a desired information resource

[3] A *Semantic Web document* is a web page that serializes an RDF graph using one of the recommended RDF syntax languages, i.e., RDF/XML, N-Triples or N3.

One advantage of the Semantic Web is that people can collaboratively create ontologies and build common vocabulary without centralized control. One building block of Semantic Web ontologies is a **Semantic Web Term** (SWT)[4], which plays the role of a word in natural languages. SWTs bridge RDF statements with formal semantics defined in RDF(S) and OWL, and are intended to be reused as universal symbols.

We call an SWD that defines a significant number of SWTs a **Semantic Web Ontology**(SWO) to distinguish it from documents that mostly populating and/or asserting class instances[5]. The Semantic Web depends on three "meta ontologies" (RDF, RDFS and OWL) and, according to Swoogle [1], thousands of additional ones developed by institutions (e.g., CYC, WordNet, DC[6], FOAF[7], and RSS) and individuals.

These ontologies often overlap by defining terms on similar or the same concepts. For example, Swoogle finds over 300 distinct SWTs that appear to stand for the 'person' concept. This raises interesting issues in finding and comparing Semantic Web ontologies for knowledge sharing. For example, how can an agent find the most popular domain ontology (currently FOAF is the best choice) to publish a personal profile?

Conventional web navigation and ranking models are not suitable for the Semantic Web for two main reasons: (i) they do not differentiate SWDs from the overwhelming number of other web pages; and (ii) they do not parse and use the internal structure of SWD and the external semantic links among SWDs. Hence, even Google, one of the best web search engines, can sometimes perform poorly in finding ontologies. For example, the FOAF ontology (the most used one for describing a person) is not among the first ten results when we search Google using the phrase "person ontology"[8].

Although we are familiar with surfing on the Web, navigating the Semantic Web is quite different. We have developed a Semantic Web navigation model based on how knowledge is published and accessed. To publish content, information providers need to obtain appropriate domain ontologies by reusing existing ones and/or creating new ones, and then use them to create instances and make assertions. When accessing knowledge, consumers need to search for instance data and pursue corresponding ontologies to fully understand the knowledge encoded. Meanwhile, the navigation model should also acknowledge the context – the Web, which physically hosts RDF graphs in SWDs. Most existing navigation tools (e.g., HyperDAML[9] and Swoop[10]) employ the URL semantics of the URIref to a RDF resource; however, they cannot answer questions like "find instances of a given class" or "list all URIs using the same local name *person*" due to the limited number of explicit links.

The navigation model supports ranking the 'data quality' [2] of Semantic Web knowledge in terms of common case importance. In particular, this paper focuses on

[4] A *Semantic Web term* is an RDF resource that represents an instance of rdfs:Class (or rdf:Property) and can be universally referenced by its URI reference (URIref).

[5] Since virtually all documents will contain some definitions and instances, the classification must either be a fuzzy one or depend on a heuristic threshold.

[6] Dublin Core Element Set 1.1,http://purl.org/dc/elements/1.1/.

[7] Friend Of A Friend ontology, http://xmlns.com/foaf/0.1/.

[8] This example is not intended to undermine Google's value; instead, we argue that the Semantic Web is quite different from the Web and needs its own navigation and ranking models.

[9] http://www.daml.org/2001/04/hyperdaml/

[10] http://www.mindswap.org/2004/SWOOP/

ranking ontologies at various levels of granularity to promote reusing ontologies. Ranking ontologies at the document level has been widely studied since most ontologies are published through SWOs. Its common approaches include link-analysis [3, 1] and semantic-content-analysis [4]. Document level ontology ranking, however, is not enough. For example, foaf:Person and dc:creator together can describe the author of a web page, and an ontology containing both of the concepts might not be as good as the combination of FOAF and DC. Hence, a finer level of granularity (i.e., ranking at SWT level) is needed especially to encode knowledge using popular terms from multiple ontologies[11], but is seldom investigated in literature. Besides ranking individual SWTs, agents may also rank inter-term relations (e.g., how frequently a property has been used to modify the instances of a class). Such an ontology ranking approach is a special case of ranking sub-graphs of an RDF graph [5, 6].

The remainder of this paper is structured as follows: Section 2 reviews the test-bed (the Swoogle Semantic Web search engine) and related works on navigating and ranking Semantic Web knowledge. Section 3 introduces the novel Semantic Web navigation model, which enriches navigation paths and captures surfing behaviors on the Semantic Web on the Web. Sections 4 and 5 describe and evaluate mechanisms for ranking ontologies at different levels of granularity, namely document, term and sub-graph. Section 6 concludes that effective navigation support and ranking mechanisms are critical to both the emergence of common ontologies and the growth of the Semantic Web on the Web.

2 Background and Related Work

2.1 Swoogle

The Swoogle [1] search engine discovers, indexes, and analyzes Semantic Web documents published on the Web and provides agents with various kinds of search services. Its architecture, shown in Figure 1, is comprised of four components.

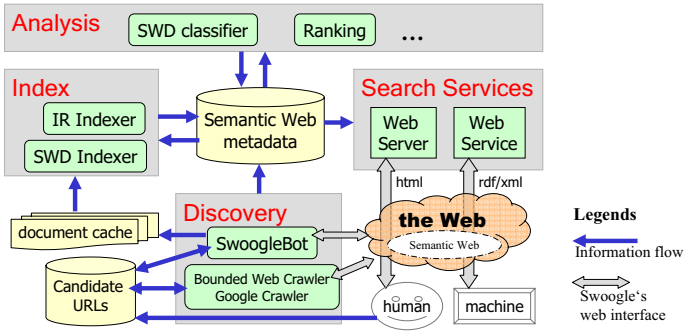

Fig. 1. Swoogle's architecture involves four major components.

[11] Importing part of ontologies is especially helpful when using large upper ontologies like CYC.

– The **Discovery** component collects candidate URLs to find and cache SWDs using four mechanisms: (i) submitted URLs of SWDs and sites; (ii) a web crawler that explores promising sites; (iii) a customized meta-crawler that discovers likely URLs using conventional search engines; and (iv) the SwoogleBot Semantic Web crawler which validates and analyses SWDs to produce new candidates.
– The **Indexing** component analyzes the discovered SWDs and generates the bulk of Swoogle's metadata about the Semantic Web. The metadata not only characterizes the features associated with individual SWDs and SWTs, but also tracks the relations among them, e.g., "how SWDs use/define/populate a given SWT" and "how two SWTs are associated by instantiating 'rdfs:domain' relation".
– The **Analysis** component analyzes the generated metadata and hosts the modular ranking mechanisms.
– The **Services** module provides search services to agents, allowing them to access the metadata and navigate the Semantic Web. It is highlighted by the "Swoogle Search" service that searches SWDs using constraints on URLs, the SWTs being used or defined, etc.; and the "Ontology Dictionary" service that searches ontologies at the term level and offers more navigational paths.

2.2 Related Work and Motivation

Random Surfing Model and PageRank. The random surfing model underlying the PageRank [7] algorithm has been widely accepted as the navigation model for the Web. In this model, the surfer begins by jumping to a random URL. After visiting a page, he either (i) with probability d[12] randomly chooses a link from the page to follow to a new page; or (ii) with probability $1 - d$ jumps to another random URL. This model is essentially a simple random walk modeled by a Markov chain. Based on this surfing model, the basic PageRank algorithm computes the rank (indicating popularity rather than relevance) for each web page by iteratively propagating the rank until convergence.

Variations of PageRank. The basic PageRank algorithm is limited by its assumptions and relaxing them has resulted in several extensions. In Topic-Sensitive PageRank [8], documents are accessed non-uniformly according to their topics. For Weighted PageRank extensions [9–11], links are followed non-uniformly according to their popularity. Several link-semantics-aware extensions [12, 13] recognize links with different meanings and compute a PageRank weighted by the link semantics.

Navigating the Semantic Web. Navigating the Semantic Web is quite different from navigating the conventional Web. It is currently supported by tools such as browsers (e.g., HyperDAML and Swoop), ontology libraries (e.g., DAML ontology library[13] and SchemaWeb[14]), search engines (e.g., Ontaria[15] and Swoogle), and crawlers (e.g., scutter[16] and SwoogleBot). Most tools only capture navigational paths based on the seman-

[12] d is usually a constant except in personalized ranking.
[13] http://www.daml.org/ontologies/
[14] http://www.schemaweb.info/
[15] http://www.w3.org/2004/ontaria/
[16] http://rdfweb.org/topic/Scutter

tics of URIref. Swoogle, however, supports effective navigation by providing additional navigational paths among SWDs and SWTs.

Ranking Semantic Web knowledge. Ranking knowledge can be considered as a problem of evaluating *data quality* [2, 14] which focuses on *data product quality* [15]. It has been studied at various levels of granularity in Semantic Web and database literature.

- Ranking Semantic Web ontologies at the document level has been studied using both content analysis [16, 4] and link-structure-based analysis [3, 1].
- Ranking knowledge at the instance or object level has been investigated by both database and Semantic Web researchers, including ranking elements in XML documents [17]; ranking objects in databases [18] or the Web [19, 11]; and ranking relevant class-instances in domain specific RDF database [20].
- Ranking knowledge at a sub-graph level has been studied using ontology-based content analysis [5, 21, 6] in the context of ranking query results in the Semantic Web, and using context-based trust computation [22, 23].

Ranking Semantic Web ontologies has remained at the document level even though other granularity levels are applicable. For example, SWTs are a special kind of class instances and should be ranked differently from normal instances. Doing so enables a retrieval system to find a set of SWTs drawn from more than one ontologies to cover a collection of target concepts.

Most link-analysis-based approaches have focused on either a particular domain (e.g., bibliographic data) or a small set of SWOs. Swoogle is unique in its ambition to discover and index a substantial fraction of the published SWDs on the Web (currently over 7×10^5 SWDs of which about 1% are SWOs).

3 Semantic Web Navigation Model

In this paper, we consider the Semantic Web materialized on the Web. To navigate such a Semantic Web, a user cannot simply rely on the URL semantics of URIref due to three main reasons: (i) the namespace of a URIref at best points to an SWO, but there are no reverse links pointing back; (ii) although rdfs:seeAlso has been widely used to interconnect SWDs in FOAF based applications, it seldom works in other SWDs; (iii) owl:imports does interlink ontologies, but such relations are rare since ontologies are usually independently developed and distributed. In addition, many practical issues should be addressed in web-scale Semantic Web data access, such as "how two reach the SWDs which are not linked by any other SWDs" and "what if the namespace of a URIref is not an SWD". It is notable that the intended users of this navigation model are both software agents, who usually search SWDs for external knowledge and then retrieve SWOs to fully understand SWDs, and Semantic Web researchers, who mainly search SWTs and SWOs for publishing their knowledge.

3.1 Overview

The navigation model is specialized for publishing and accessing Semantic Web knowledge as shown in Figure 2. Users can jump into the Semantic Web using conventional Web search (e.g., Google and Yahoo) or Semantic Web search (e.g., Swoogle). Users can also navigate the Semantic Web within or across the Web and RDF graph via seven groups of navigational paths. An example is shown in Figure 3.

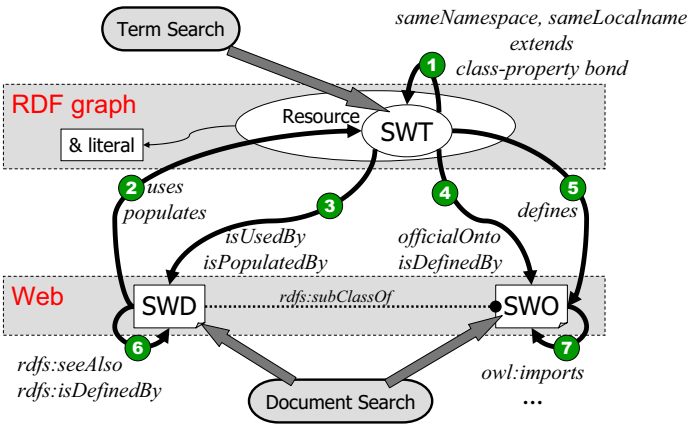

Fig. 2. The Semantic Web navigation model.
The block arrows link search services to the Semantic Web. Paths 2 and 5 are straightforward since SWTs are referenced by SWDs/SWOs. Paths 6, 7 and part of 4 are supported by most existing RDF browsers. Paths 1, 3 and the rest of 4 require global view of the Semantic Web on the Web, and are currently only supported by Swoogle metadata.

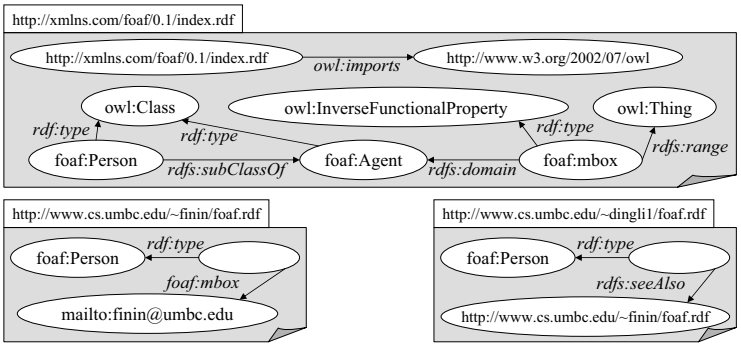

Fig. 3. A navigation use-case.
A user can use Swoogle term search to find SWTs having local name 'Person'. If she picks SWT *foaf:Person*, she can jump to the corresponding SWO http://xmlns.com/foaf/0.1/index.rdf by following path 4 via *isDefinedBy*, jump to an SWT *foaf:mbox* by following path 1 via *sameNamespace*, or jump to another SWD http://www.cs.umbc.edu/ dingli1/foaf.rdf by following path 3 via *isPopulatedBy*. From the FOAF SWO, she can pursue OWL ontology by following path 7 via *owl:imports*, jump to SWT *rdfs:domain* by following path 2 via *populates*, or jump to SWT *foaf:Agent* by following path 5 via *defines*. For the SWD to the right, she can jump to another SWD http://www.cs.umbc.edu/ finin/foaf.rdf by following path 6 via *rdfs:seeAlso*.

In addition to conventional document search using document properties and/or bag-of-word model, Swoogle lets users locate Semantic Web knowledge using navigational paths, e.g., "a personal profile ontology can be located if it *defines* SWTs like 'person', 'email' and 'homepage' ". We detail three groups of navigational paths as follows.

3.2 Paths between SWTs

We concentrate on three of the many interesting navigational paths between SWTs grouped by path 1 in Figure 2 as follows.

1. **sameNamespace** and **sameLocalname**. linking SWTs sharing the same namespace is needed because they are not necessarily defined in the document pointed by the namespace. Linking SWTs sharing the same local name is needed to find alternative SWTs because the local name part of an SWT usually conveys its semantics.
2. **extends**. An SWT $t1$ *extends* another SWT $t2$ when either (i) there exists a triple ($t1$, P, $t2$) where P (e.g., *rdfs:subClassOf*, *owl:inverseOf* and *owl:complementOf*) connects two classes (or two properties), or (ii)there exists a triple ($t1$, P, $LIST$) where P (e.g., *owl:unionOf*) connects a class $t1$ to a *rdf:List* $LIST$ which has another class $t2$ as a non-nil member. For example, in Figure 3, *foaf:Agent* is extended by *foaf:Person* because it is closer to the concept 'person' and its *mobx* property can be inherit. The *extends* relation is a good indicator for the importance of term because it implies that the term being extended is commonly accepted and well-defined but too general for instantiating the intended concept.
3. **class-property bond**. Although classes and their attributes have been tightly bonded in frame-based systems, the connections between classes and properties are loose in the Semantic Web. For example, Dublin Core defines widely used properties without specifying their domains and ranges. Swoogle links from a class to its instance properties (i.e., class-property bond) using two sources: (i) rdfs:domain assertions in SWOs and (ii) instantiation of such bond in class-instances.

3.3 Paths between SWDs and SWTs

Swoogle maintains three types of navigational paths across SWDs and SWTs: (i) paths 2 and 5 in Figure 2 can be easily extracted from an SWD by analyzing the usage of SWTs; (ii) paths 3 and 4 are mainly the reverse of paths 2 and 5. Generating such paths requires the global view of the Semantic Web; and (iii) the **officialOnto** relation in path 4 links an SWT to an SWO. It is needed by software agents to locate ontologies defining the encountered SWTs in the absence of explicit import instruction.

1. Swoogle recognizes six types of binary relations between an SWT T in an SWD D as shown in Table 1. They can be further generalized to three groups namely, *defines*, *uses* and *populates*. For example, in figure 3, `http://xmlns.com/foaf/0.1/index.rdf` defines foaf:Person as class and populates rdfs:domain as property. An SWD using or populating an SWT indicates that the publisher is satisfied with the SWT's definition.

Table 1. Six types of binary relations that can hold between an SWD D and an SWT T

Relation	Condition
define-class	D has a triple (T, *rdf:type*, *MC*) where *MC* is a sub-class of *rdfs:Class*.
define-property	D has a triple (T, *rdf:type*, *MP*) where *MP* is a sub-class of *rdf:Property*.
use-class	D has a triple (_, P, T) where the range of P is a sub-class of *rdfs:Class*, or D has a triple (T, P, _) where the domain of P is a sub-class of *rdfs:Class*.
use-property	D has a triple (_, P, T) where the range of P is a sub-class of *rdf:Property*, or D has a triple (T, P, _) where the domain of P is a sub-class of *rdf:Property*.
populate-class	D has a triple (_, rdf:type, T).
populate-property	When D has a triple (_, T, _).

2. Swoogle tracks the "official ontology" of an SWT T using heuristics listed in Table 2. The 'percent' column shows the percentage that the heuristics has been successfully applied. It is notable that heuristics 2 and 3 help find some important official ontologies of DC and FOAF even though they have only improved the performance from 59% to 62.8%.

Table 2. Heuristics for finding official ontologies, and their performance on 4508 namespaces.

Type	Percent
The namespace of T;	59%
the URL of an ontology which is redirected from T's namespace (e.g., `http://purl.org/dc/elements/1.1/` is redirected to `http://dublincore.org/2003/03/24/dces`);	0.4%
the URL of an ontology which has T' namespace as its absolute path, and it is the only one that matches this criteria (e.g., `http://xmlns.com/foaf/0.1/index.rdf` is the official ontology of `http://xmlns.com/foaf/0.1/`);	3.4%
N/A, cannot decide	37.2%

3.4 Paths between SWDs

Swoogle also supports well-known navigational paths between SWDs.

1. Although not defined explicitly, the triples populating properties *rdfs:isDefinedBy* and *rdfs:seeAlso* are widely used in linking to web pages or even SWDs. In practice, many RDF crawlers use *rdfs:seeAlso* to discover SWDs.
2. Instances of *owl:OntologyProperty* is explicitly defined to associate two SWOs, and owl:imports is frequently instantiated far more than the others. Therefore, Swoogle indexes the usage of the **imports**[17] relation.

[17] An SWO $D1$ *imports* another $D2$ when there is a triple in $D1$ in form of ($D1$, owl:imports, $D2$), and so does *daml:imports*. This relation shows the dependency between ontologies and is complemented by "officalOnto" relation.

3. Inspired by RDF test-case ontology[18], we have developed a class *wob:RDFDocument* (which asserts that a resource is an SWD) to support explicit 'hyperlinks' in the Semantic Web. A consequent idea is RDF sitemap which let website publish their SWDs through a special index file[19].

4 Ranking Semantic Web Documents

Since RDF graphs are usually accessed at the document level, we simplify the Semantic Web navigation model by generalizing navigational paths into three types of document level paths (see below) and then applying link analysis based ranking methods with 'rational' surfing behavior.

- An **extension (EX)** relation holds between two SWDs when one defines a term using terms defined by another. EX generalizes the *defines SWT-SWD* relations, the *extends SWT-SWT* relations, and the officialOnto *SWT-SWD* relation. For example, an SWD $d1$ *EX* another SWD $d2$ when $d1$ defines a class $t1$, which is the subclass of a class$t2$, and $t2$'s official ontology is $d2$.
- A **use-term (TM)** relation holds between two SWDs when one uses a term defined by another. TM generalizes the *uses* and *populates SWT-SWD* relations, and the officialOnto *SWT-SWD* relation. For example, an SWD $d1$ *TM* another SWD $d2$ when $d1$ uses a resource t as class, and t's official ontology is $d2$.
- An **import (IM)** relation holds when one SWD *imports*, directly or transitively, another SWD, and it corresponds to the imports *SWD-SWD* relation.

4.1 Rational Surfer Model and OntoRank

Swoogle's *OntoRank* is based on the *rational surfer model* which emulates an agent's navigation behavior at the document level. Like the random surfer model, an agent either follows a link in an SWD to another or jumps to a new random SWD with a constant probability $1 - d$. It is 'rational' because it emulates agents' navigation on the Semantic Web, i.e., agents follow links in a SWD with non-uniform probability according to link semantics. When encountering an SWD α, agents will (transitively) import the "official" ontologies that define the classes and properties referenced by α.

Let $link(\alpha, l, \beta)$ be the semantic link from an SWD α to another SWD β with tag l; $linkto(\alpha)$ be a set of SWDs link directly to the SWD α; $weight(l)$ be a user specified navigation preference on semantic links with type l, i.e., TM and EX; $OTC(\alpha)$ be a set of SWDs that (transitively) IM or EX α as ontology; $f(x, y)$ and $wPR(x)$ be two intermediate functions.

OntoRank is computed in two steps: (i) iteratively compute the rank, $wPR(\alpha)$, of each SWD α until it converges (equations 1 and 2); and (ii) transitively pass an SWD's rank to all ontologies it imported (equation 3).

[18] http://www.w3.org/2000/10/rdf-tests/rdfcore/testSchema
[19] http://swoogle.umbc.edu/site.php

$$wPR(\alpha) = (1-d) + d \sum_{x \in linkto(\alpha)} \frac{wPR(x) \times f(x,\alpha)}{\sum\limits_{link(x,\_,y)} f(x,y)} \quad (1)$$

$$f(x,\alpha) = \sum_{link(x,l,\alpha)} weight(l) \quad (2)$$

$$OntoRank(\alpha) = wPR(\alpha) + \sum_{x \in OTC(\alpha)} wPR(x) \quad (3)$$

4.2 Evaluation: OntoRank vs PageRank

OntoRank is evaluated on a real dataset *DS-APRIL* collected by Swoogle by April 2005. DS-APRIL contains 330K SWDs (1.5% are SWOs, 24% are FOAF documents and 60% are RSS documents) and interlink by 200K document level relations.

The first experiment compares the performance between PageRank and OntoRank in boosting the rank of SWOs among SWDs, i.e., ranking SWOs higher than normal SWDs. In this experiment, we first compute both ranks for SWDs in DS-APRIL[20]; and then ten popular local-names (according to Swoogle's statistics) were selected as the keywords for Swoogle's document search. The same search result for each query is ordered by both PageRank and OntoRank respectively. We compared the number of *strict SWO* (see definition 1) in the first 20 results in either order. Table 3 shows an average 40% improvement of OntoRank over PageRank.

Table 3. OntoRank finds more ontologies in each of the 10 queries

Query	**C1:# SWOs by OntoRank**	**C2:# SWOs by PageRank**	**Difference (C1-C2)/C2**
name	9	6	50.00%
person	10	7	42.86%
title	13	12	8.33%
location	12	6	100.00%
description	11	10	10.00%
date	14	10	40.00%
type	13	11	18.18%
country	9	4	125.00%
address	11	8	37.50%
organization	9	5	80.00%
Average	11.1	7.9	40.51%

[20] Note this PageRank is computed on the same dataset as OntoRank, which is a preprocessed web of SWDs where no simply hyperlinks but only semantic links are considered.

Definition 1. ontology ratio
The ontology ratio *of an SWD refers to is the fraction of its class-instances being recognized as classes and properties. It is used to identify SWOs among SWDs. For example, given an SWD defining a class "Color" and populating the class with three class-instances namely, 'blue', 'green' and 'red', its ontology ratio is 25% since only one out of the four is defined as class. A document with a high ontology ratio indicates a preference for adding term definition rather than populating existing terms. According to Swoogle, an SWD is an ontology document if it has defined at least one term, and it is called a* **strict SWO** *if its ontology ratio exceeds 0.8.*

The second experiment studies the best ranked SWDs using both ranking methods. In table 4, RDFS schema clearly ranks first according to both OntoRank and PageRank. OWL ranks higher than RDF because it is referred to by many popular ontologies. DC and FOAF ontologies rank 4th and 5th by PageRank due to their many instance documents but rank lower by OntoRank due to their narrow domain and fewer references by other ontologies. An interesting case is the web of trust (WOT) ontology which PageRank ranks only 29th since our data set only contains 280 FOAF documents referencing it directly. OntoRank ranks it at 8 since it is referenced by the FOAF ontology, greatly increasing its visibility. We are not expecting OntoRank to be completely different from PageRank since it is a variation of PageRank. OntoRank is intended to expose more ontologies which are important to Semantic Web users in understanding term definition.

Table 4. Top 10 SWDs according to OntoRank and their PageRank

URL of Ontology	Ontology Ratio	OntoRank	PageRank
http://www.w3.org/2000/01/rdf-schema	94%	1	1
http://www.w3.org/2002/07/owl	86%	2	5
http://www.w3.org/1999/02/22-rdf-syntax-ns	81%	3	6
http://purl.org/dc/elements/1.1	100%	4	3
http://purl.org/rss/1.0/schema.rdf	100%	5	2
http://www.w3.org/2003/01/geo/wgs84_pos	100%	6	10
http://xmlns.com/foaf/0.1/index.rdf	84%	7	4
http://xmlns.com/wot/0.1/index.rdf	100%	8	29
http://www.w3.org/2003/06/sw-vocab-status/ns	75%	9	7
http://www.daml.org/2001/03/daml+oil	96%	10	11

5 Ranking for Ontology Dictionary

Ranking ontologies at the term level is also important because SWTs defined in the same SWO are instantiated in quite different frequency. For example, owl:versionInfo is far less used than owl:Class. Users, therefore, may want to partition ontologies and then import a part of an SWO [24, 25]. In addition, users often use SWTs from multiple ontologies together, e.g., *rdfs:seeAlso* and *dc:title* have been frequently used modifying the instances of *foaf:Person*.

These observations lead to the "Do It Yourself" strategy i.e., users can customize ontologies by assembling relevant terms from popular ontologies without importing them completely. To this end, Swoogle's *Ontology Dictionary* helps users to find relevant terms ranked by their popularity, and supports a simple procedure *CONSTRUCT-ONTO* for publishing knowledge using class-instances.

```
CONSTRUCT-ONTO
  1. find an appropriate class C
  2. find popular properties whose domain is C
  3. go back to step 1 if another class is needed
```

5.1 Ranking Semantic Web Terms

Swoogle uses TermRank to sort SWTs by their popularity, which can be simply measured by the number of SWDs using/populating an SWT. This naive approach, however, ignores users' rational behavior in accessing SWDs, i.e., users access SWDs with non-uniform probability. Therefore, *TermRank* is computed by totaling each SWD's contribution (equation 4). For each SWD α, its contribution to each of its SWTs is computed by splitting its OntoRank proportional to SWTs' weight $TWeight(\alpha, t)$, which indicates the probability a user will access t when browsing α. $TWegiht$ is the product of $cnt\_uses(\alpha, t)$ - t's popularity within α measured by the number of occurrence of t in α and $|\{\alpha|uses(\alpha, t)\}|$ – t's importance in the Semantic Web measured by the number of SWDs containing t (see equation 5).

$$TermRank(t) = \sum_{uses(\alpha,t)} \frac{OntoRank(\alpha) \times TWeight(\alpha,t)}{\sum_{uses(\alpha,x)} TWeight(\alpha,x)} \tag{4}$$

$$TWeight(\alpha, t) = cnt\_uses(\alpha, t) \times |\{\alpha|uses(\alpha, t)\}| \tag{5}$$

Table 5 lists top ten classes in *DS-APRIL* having 'person' as the local name ordered by TermRank. For each class, *pop(swd)* refers to the number of SWDs populating it; *pop(i)* refers to the number of its instances; and *def(swd)* refers to the number of SWDs defining it. Not surprisingly, *foaf:Person* is number one. The sixth term is a common mis-typing of the first one, so it has been well populated without being defined. The ninth term has apparently made the list by virtue of the high OntoRank score of the SWO that defines it.

Table 6 lists top ten SWTs in Swoogle's Ontology Dictionary. The *type* of an SWT is either 'p' for property or 'c' for class. *rdfs:comment* is ranked higher than *dc:title* even though the latter is better populated because the former is referenced by many important SWDs. Properties are ranked higher than classes since they are less domain specific.

5.2 Ranking Class-Property Bonds

A more specific issue directly related to step 2 in *CONSTRUCT-ONTO* is ranking *class-property bonds* (see definition 2), which helps users choose the most popular properties for a class when they are publishing data with the desire of maximizing the data's

Table 5. Top ten classes with 'person' as the local name ordered by Swoogle's TermRank

TermRank	Resource URI	pop(swd)	pop(i)	def(swd)
1	http://xmlns.com/foaf/0.1/Person	74589	1260759	17
2	http://xmlns.com/wordnet/1.6/Person	2658	785133	80
3	http://www.aktors.org/ontology/portal#Person	267	3517	6
4	ns1:Person [1]	257	935	1
5	ns2:Person [2]	277	398	1
6	http://xmlns.com/foaf/0.1/person	217	5607	0
7	http://www.amico.org/vocab#Person	90	90	1
8	http://www.ontoweb.org/ontology/1#Person	32	522	2
9	ns3:Person [3]	0	0	1
10	http://description.org/schema/Person	10	10	0

[1] ns1 - http://www.w3.org/2000/10/swap/pim/contact#
[2] ns2 - http://www.iwi-iuk.org/material/RDF/1.1/Schema/Class/mn#
[3] ns3 - http://ebiquity.umbc.edu/v2.1/ontology/person.owl#

Table 6. Top ten terms order by TermRank

TermRank	SWT	type	pop(swd)	pop(i)
1	rdf:type	p	334810	8174201
2	dc:description	p	60427	918644
3	rdfs:label	p	12795	197079
4	rdfs:comment	p	4626	137267
5	dc:title	p	60229	1452612
6	rdf:Property	c	4117	52445
7	dcterms:modified	p	11881	25321
8	rdfs:seeAlso	p	55985	1167786
9	dc:language	p	149878	225600
10	dc:type	p	9461	54676

visibility. For example, when publishing an instance of *foaf:Person*, we might always supply a triple that populates the most common property *foaf:mbox_sha1sum*.

Definition 2. *A* **class-property bond (c-p bond)** *refers to an* rdfs:domain *relation between property and class. While c-p bonds can be specified in ontologies in various ways, e.g., direct association (*rdfs:domain) and class-inheritance; we are interested in finding c-p bonds in class instances characterized by the two-triple graph pattern: $(\_x, rdf:type, class), (\_x, property, \_)$.

To rank c-p bonds, we cannot simply rely on the definition from ontologies because that does not show how well a c-p bond has been adopted in practice. We evaluate c-p bonds, therefore, by ranking the subgraph that instantiates c-p bonds, e.g., the number instance of foaf:person modified by foaf:name. In DS-APRIL, the five highest ranked properties (by the number of SWDs instantiated c-p bond) of *foaf:Person* are (i) foaf:mbox_sha1sum (67,136 SWDs), (ii) foaf:nick (62,266), (iii) foaf:weblog (54,341), (iv) rdfs:seeAlso (47,228), and (v) foaf:name (46,590).

6 Conclusions and Future Work

Swoogle supports two primary use cases: helping human knowledge engineers find ontologies and terms and serving agents and tools seeking knowledge and data. While no formal evaluation has yet been done, we offer some observations that address how well Swoogle meets its goals and informally compare it to the alternatives.

Swoogle's web-based service has been available since Spring 2004 and has received several million hits, supporting hundreds of regular users and thousands of casual ones. Swoogle continuously discovers online SWDs and thus maintains a global view of the public Semantic Web. The results reported here are based on a dataset (DS-APRIL) of over 330,000 SWDs and 4,000 SWOs, about half the size of the current collection. Swoogle has found many more SWDs, most of which are FOAF or RSS documents, that are excluded from the database to make Swoogle's dataset balanced and interesting. Swoogle's ability to search content at various granularity levels and its ranking mechanisms are novel and promote the emergence of consensus ontologies.

There are three alternatives to Swoogle that can be used to find knowledge on the Semantic Web: conventional search engines, Semantic Web repositories, and specialized RDF data collections. Some conventional search engines index RDF documents and can be used to find SWDs and SWTs. However, none understands the content being indexed, recognizes terms as links, or even correctly parses all RDF encodings. Any ranking done by such systems ignores links between SWDs and their corresponding semantic relationships. Some useful SWD repositories are available (e.g., those at www.schemaweb.info and rdfdata.org) but require manual submission and have limited scope. Several crawler-based systems exist that are specialized to particular kinds of RDF (e.g., FOAF, RSS, DOAP, Creative Commons), but their scope and services are restricted. Intellidimention has an experimental crawler based system [21] similar to Swoogle but with abridged coverage.

A formal evaluation of Swoogle's performance on finding and ranking SWDs and SWTs would be based, in part, on measuring the precision and recall for a set of queries against human judgments. This would allow us to compare Swoogle's performance to other systems, to evaluate different ranking algorithms and to evaluate the impact of doing more or less inference. While we intend to carry out such an evaluation, it requires careful design and significant labor to acquire the necessary human evaluations. User studies through questionnaires or surveys on Swoogle ranking results are planned to provide a subjective reference.

By enlarging the test dataset and compensating for biases due to the predominance of FOAF and RSS documents, we expect to refine our evaluation of Swoogle's navigation model and ranking algorithms. We are also improving the ranking algorithms without generalizing the navigation model, motivated by the the success of XML object-level ranking [17, 11]. We are extending class-property bond ranking to a more general issue – tracking the provenance of and ranking arbitrary RDF sub-graphs [26]. This can be used to resolve, for example, a case where multiple RDF triples claim different values for a person's homepage (whose cardinality constraint is one).

[21] http://www.semanticwebsearch.com/

References

1. Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C., Sachs, J.: Swoogle: A search and metadata engine for the semantic web. In: CIKM'04. (2004)
2. Wang, R., Storey, V., Firth, C.: A framework for analysis of data quality research. IEEE Transactions on Knowledge and Data Engineering **7** (1995) 623–639
3. Patel, C., Supekar, K., Lee, Y., Park, E.K.: OntoKhoj: a semantic web portal for ontology searching, ranking and classification. In: WIDM'03. (2003) 58–61
4. Alani, H., Brewster, C.: Ontology ranking based on the analysis of concept structures. In: Proc. of the 3rd International Conference on Knowledge Capture (K-Cap). (2005)
5. Stojanovic, N., Studer, R., Stojanovic, L.: An approach for the ranking of query results in the semantic web. In: ISWC'03. (2003)
6. Anyanwu, K., Maduko, A., Sheth, A.: Semrank: Ranking complex relationship search results on the semantic web. In: WWW'05. (2005) 117 –127
7. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University (1998)
8. Haveliwala, T.H.: Topic-sensitive pagerank. In: WWW'02. (2002) 517–526
9. Jeh, G., Widom, J.: Scaling personalized web search. In: WWW '03. (2003) 271–279
10. Xing, W., Ghorbani, A.A.: Weighted pagerank algorithm. In: Proc. of the 2nd Annual Conference on Communication Networks and Services Research. (2004) 305–314
11. Nie, Z., Zhang, Y., Wen, J.R., Ma, W.Y.: Object-level ranking: Bringing order to web objects. In: WWW'05. (2005) 567–574
12. Zhuge, H., Zheng, L.: Ranking semantic-linked network. In: WWW'03 Posters. (2003)
13. Baeza-Yates, R., Davis, E.: Web page ranking using link attributes. In: WWW'04 Posters. (2004) 328–329
14. Wand, Y., Wang, R.Y.: Anchoring data quality dimensions in ontological foundations. Communications of the ACM **39** (1996) 86–95
15. Kanh, B.K., Strong, D.M., Wang, R.Y.: Information quality benchmarks: Product and service performance. Communications of the ACM **45** (2002) 184–192
16. Supekar, K., Patel, C., Lee, Y.: Characterizing quality of knowledge on semantic web. In: Proc. of 7th International Florida Artificial Intelligence Research Society Conf. (2002)
17. Guo, L., Shao, F., Botev, C., Shanmugasundaram, J.: XRANK: ranked keyword search over XML documents. In: SIGMOD'03. (2003) 16–27
18. Balmin, A., Hristidis, V., Papakonstantinou, Y.: ObjectRank: Authority-based keyword search in databases. In: VLDB'04. (2004) 564 – 575
19. Xi, W., Zhang, B., Chen, Z., Lu, Y., Yan, S., Ma, W.Y., Fox, E.A.: Link fusion: A unified link analysis framework for multi-type interrelated data objects. In: WWW'04. (2004) 319–327
20. Rocha, C., Schwabe, D., Aragao, M.P.: A hybrid approach for searching in the semantic web. In: WWW'04. (2004) 374–383
21. Aleman-Meza, B., Halaschek, C., Arpinar, I.B., Sheth, A.: Context-aware semantic association ranking. In: SWDB'03. (2003) 33–50
22. Bizer, C.: Semantic web trust and security resource guide. (http://www.wiwiss.fu-berlin.de/suhl/bizer/SWTSGuide/ (last accessed 08-11-05))
23. Ding, L., Kolari, P., Finin, T., Joshi, A., Peng, Y., Yesha, Y.: On homeland security and the semantic web: A provenance and trust aware inference framework. In: Proceedings of the AAAI Spring Symposium on AI Technologies for Homeland Security. (2005)
24. Volz, R., Oberle, D., Maedche, A.: Towards a modularized semantic web. In: Proceedings of the ECAI'02 Workshop on Ontologies and Semantic Interoperability. (2002)
25. Grau, B.C., Parsia, B., Sirin, E.: Working with multiple ontologies on the semantic web. In: ISWC'04. (2004)
26. Ding, L., Finin, T., Peng, Y., da Silva, P.P., McGuinness, D.L.: Tracking rdf graph provenance using rdf molecules. Technical Report TR-05-06, UMBC (2005)