Finding and Ranking Knowledge on the Semantic Web *

Li Ding[1], Rong Pan[1], Tim Finin[1], Anupam Joshi[1], Yun Peng[1], and Pranam Kolari[1]

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 is a system that helps knowledge engineers and software agents find knowledge on the web encoded in the semantic web languages RDF and OWL. Based on the search mechanisms provided in the previous version, we propose a novel semantic web navigation model and refine mechanisms for ranking the semantic web at various granularities. Although the semantic web is materialized on the Web, it is hard to navigate within the semantic web since few explicit "hyperlinks" are available besides a URIref's namespace or owl:import semantic. Hence we propose a navigation model that characterizes users' navigational behavior (e.g. surfing from an ontology to one class C defined in it, and then to the RDF documents that populate C or the other resources that help defining this class) within the semantic web and implement it in Swoogle's "Ontology Dictionary". Based on this navigation model and the metadata collected in Swoogle, we have developed algorithms for ranking objects in the semantic web at various levels of granularity including semantic web document (SWD) level, resource level (e.g., RDF class or property) and triple level (e.g. interesting RDF graph pattern). Ranking SWDs, inspired by the Google's PageRank, emulates an "rational" agent acquiring knowledge on the semantic web using the hyperlinks provided by our "semantic web navigation model" at document level. Ranking individual terms extends ranking to a finer granularity. For example, from the hundreds of RDF terms denoting the concept of a person, the question of "which are most widely used?" is answered by term ranking. Finally, we introduce the notion of ranking facts (e.g., RDF triples) such as the rdfs:domain relation between a class and a property using provenance based heuristics. These ranking mechanisms, if being used in filtering ontologies, could help the emergence of consensus ontologies. Experiments show that the Swoogle search engine using "semantic ranking" outperforms Google in evaluating the importance of ontologies.

1 Introduction

The semantic web aims to be a distributed information system for software agents to publish, discover, and share knowledge. Its current realization is as a collection of RDF graphs, each of which is serialized in a web document. We refer to these documents as *semantic web documents*.

* 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.

Definition 1. *A* **semantic web document (SWD)** *is a web document that serializes an RDF graph using recommended RDF syntax languages, i.e. RDF/XML, N-Triples or N3. A* **semantic web term (SWT)** *is a named RDF resource (i.e. having a URI) which is defined as an instance of* rdfs:Class *(or* rdf:Property*). An SWD is called a* **semantic web ontology (SWO)** *if it has defined some semantic web terms.*

One advantage brought about by the semantic web is that people can create ontologies collaboratively without centralized control. This feature results in thousands of ontologies in the web as discovered by Swoogle [1]. Besides the well known and institution-backed ontologies such as CYC, SUMO, WordNet, Dublin Core, FOAF, and RSS, there are many ontologies developed by individuals. These ontologies often overlap, for example, the concept "person" is defined by hundreds of ontologies using different namespaces. This raises some interesting issues regarding the effective use of ontologies in expressing and sharing knowledge in a distributed environment such as the web. For example, a user would like to use the most popular domain ontologies to share her knowledge, but how can she find them?

Conventional web document navigation and ranking models are not suitable for the semantic web due to many reasons: (i) they do not differentiate SWDs from the much larger number of html documents; (ii) they ignore the semantics of the links between SWDs. Hence even Google, one of the best web search engines, can sometime perform poorly in finding ontologies. For example, the FOAF ontology, which is one of the best ontologies for describing a person, is not listed in the top ten results when we search Google using the phrase "person ontology"[1].

This paper proposes a novel navigation model that captures the relations between the unified RDF graph provided by the semantic web and its web context (the web of SWDs). Our navigation model is materialized as Swoogle's semantic web metadata database and web interface (including Swoogle Search, Document Digest and Ontology Dictionary)[2]. Based on this model, we also rank SWDs to estimate their importance in terms of how widely they are used. We also extend ranking to objects at a finer granularity in **ontology dictionary**, i.e. we rank classes/properties as well as interesting RDF graph patterns (e.g. the rdfs:domain relation between class and properties). With term level ranking, users can assemble popular terms from multiple ontologies to express knowledge in their applications without the concern of importing entire ontologies. This is especially helpful when using terms from large upper ontologies like CYC and SUMO.

Contributions This work turns out to be one of the first works that model navigation paths in the semantic web. The *semantic web navigation model* connects the web of SWDs with the unified RDF graph from the entire semantic web, and hence enrich the navigational paths in the semantic web. Another contribution of this work is the ranking mechanisms, which start a systematic investigation of "data quality"[2] issues in semantic web context at various granularities. Document is not the only object of interests; moreover, finer granularity objects, namely terms (nodes) and triples (edges)

[1] This example is not intended to undermine Google's value; instead, we argue that the semantic web is a quite different from the normal web and require its own navigation and ranking models.

[2] http://swoogle.umbc.edu/

in RDF graph, could also be ranked. These quantitative ranking mechanisms can also enable the emergence of common ontologies at document level or term level granularity.

2 Background

2.1 Swoogle

Swoogle[1] is a crawler-based indexing and retrieval system for SWDs. Though Swoogle is pretty used predominately by human users, our vision is that its software agent users will increase and dominate in the future. Swoogle discovers, digests, analyzes and indexes online SWDs so as to help agents search and navigate knowledge in the semantic web. Figure 1 shows Swoogle's architecture.

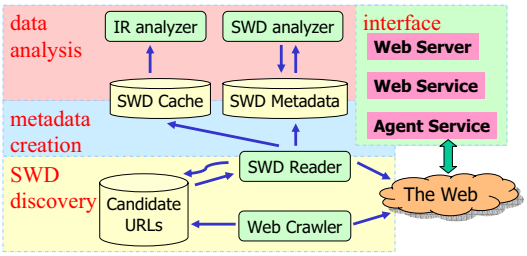

Fig. 1. Swoogle's architecture has four components that work together to discover, digest, analyze and serve up semantic web knowledge and data.

- The **discovery** component automatically discovers and revisits SWDs throughout the Web using a set of integrated web crawlers.
- The **digest** component generates metadata for individual SWDs and semantic web terms (SWTs) as well as identifies relations among them, e.g. "how one SWD references another by using an SWT defined by the latter", "which SWD uses a given SWT", and "is a class the domain of a property".
- The **analysis** component uses cached SWDs and their metadata to derive analytical reports, such as identifying ontologies among SWDs and ranking SWDs by their importance.
- The **service** component provides services to both human users and software agents through conventional web interface and SOAP-based web service interface respectively. It is highlighted by i) "Swoogle Search" service that searches SWDs by constraints on their URLs, the sites which host them, and the classes/properties used or defined by them; and ii) "Ontology Dictionary" service that indexes all discovered SWTs and provides more navigational paths across the web of SWTs and the web of SWDs.

2.2 PageRank and its Implication

PageRank, introduced by Page et al[3], is a user independent importance measurement of web pages based on the analysis of their link structures. It estimates the probability that a page is visited by a surfer performing random walks along the hyperlinks. The process of this random walk can also be modeled by a Markov chain, of which the stochastic transition matrix has a static distribution.

There have been significant extensions of the basic PageRank algorithm. Topic-Sensitive PageRank algorithm proposed by [4] pre-computes a set of ranking vectors biased by given topics to generate more accurate results for query with contexts. The Modular PageRank approach from [5] biased user's random walk to higher ranked pages, which are selected based on user's interests. The Random walk model can also be biased by link semantics: [6] characterized eight basic relations between web pages; [7] used heuristics from the html context of hyperlink. In semantic web context, [8] biased random walk to three types of semantic links (i.e. instantiation, subclass, domain/range) according RDF semantics.

3 Semantic Web Navigation Model

The semantic web is currently embedded in the Web, and the navigation path usually links across normal web documents and semantic web documents. As mentioned in RDF crawlers like Scutter[3], SWDs are implicitly linked together by either the namespace of a URIref or triples using predicates like *owl:imports* and *rdfs:seeAlso*. In fact, this situation has some inherent limitations, for examples, "how two reach the SWDs which are not linked by any other SWDs", "what if the namespace of a URIref is not an SWD", and "how to start with the class *rdf:Person* and jump to all RDF documents that define or use it as a class". It is notable that the intended users of this navigation model are mainly software agents, which prefers reading RDF documents to normal web documents. The model can also help semantic web researchers who do use Swoogle web interface frequently now.

This model is implemented by search/navigation services which are based on Swoogle's metadata. Those services enrich the navigation paths in the semantic web and enable users to navigate the semantic web at RDF graph level (where resources and literals are linked), at the web of RDF documents level (where RDF documents are linked), and most importantly across the two levels.

These enriched navigation paths bring about a novel semantic web navigation model that offers users more freedom in navigating the semantic web. Our model focuses on the relations within and across the *RDF graph* world and the *Web*. As shown in figure 2, our model adds many navigation paths besides the conventional ones referred to elsewhere in literature: (i) relations among Resources (relation 1), (ii) relations between SWDs and Resources (relation 2 to 5), and (iii) relations among SWDs (relation 6 and 7). It also shows that users can navigate into the semantic web through document search (or term search) if the URL of SWD or URIref of Resource is not known in advance.

[3] http://rdfweb.org/topic/Scutter

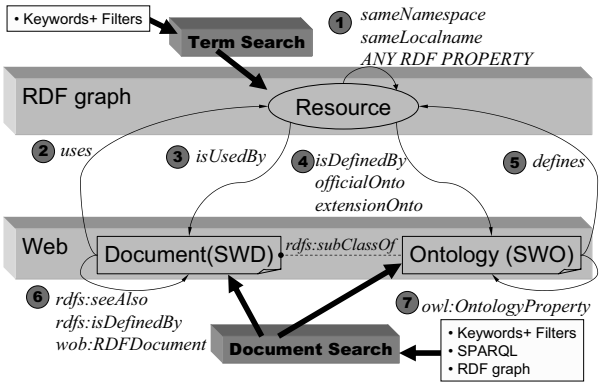

Fig. 2. Semantic web navigation model

3.1 Relations between Resources

RDF resources are linked by instances of *rdf:Property*, and we are particularly interested in links between SWTs.

A term (SWT) is extended by another usually for the following reasons: (i) it is commonly accepted (high visibility); (ii) its definition is worthy of being inhered; and (iii) it is too general for populating instances. Therefore, the direct graph of *term-extension* relation is basically taxonomy of SWTs starting at most specific ones and ending at the most general ones. It is a useful indicator for the importance of an SWT as a commonsense concept in building ontologies.

Definition 2. term-extension. *An SWT $t1$ is said extending another SWT $t2$ when any of the following conditions is met:*

- *There exists a triple ($t1$, P, $t2$), such that the domain and range of P are defined as the sub-class of* rdfs:Class *(or* rdfs:Property*).*
- *There exist a triple ($t1$, P, $LIST$), such that (i) P satisfies the above domain/range constraint, (ii) $t2$ is a non-nil member of $LIST$, and (iii) $t2$ is a class.*

Besides the instances of *rdf:Property*, users may navigate resources through the correlations between their URIrefs: (i) *sameNamespace* links SWTs under the same namespace; and (ii)*sameName* links SWTs sharing the same local name.

3.2 Relations between Documents and Resources

RDF resource is the building block of RDF graph, which is serialized in SWDs. Since SWTs are important to understand the meaning of an RDF graph, we are interested in the relations between SWTs and SWDs.

A term is mentioned in an SWD when it is a RDF node in the SWD's RDF graph. This relation can be further classified into *defines*, *uses*, and *populates*. An SWD *uses* an SWT usually for the following reasons: (i) the creator is satisfied with the term's definition; and (ii) the term is good for populate instances. *uses* is a useful indicator for the importance of a term in populating and sharing instance data.

Definition 3. *An SWD D* **defines** *an SWT T when any of the following conditions is met:*

- *D has a triple (T,* rdf:type, META*) such that* META *is a sub-class of* rdfs:Class *(or* rdf:Property*).*
- *D has a triple (T, P, _) such that the domain of P is a sub-class of* rdfs:Class *(or* rdf:Property*).*

Note that the inverse of defines *is* isDefinedBy.

Definition 4. *An SWD D* **uses** *an SWT T when any of the following conditions is met:*

- *D has a triple (_, P, T) such that the range of P is a sub-class of either* rdfs:Class *(or* rdf:Property*).*
- *D has a triple (_, T, _), i.e. T is used as the predicate of the triple.*

Note that the inverse of uses *is* isUsedBy.

Definition 5. *The* **populates** *relation is a sub-property of* uses*. An SWT T is a* **populatedClass** *in an SWD D when D has a triple (_, rdf:type, T); T is a* **populatedProperty** *in D when D has a triple (_, T, _).*

Besides deriving *TM* relation from the occurrence of an SWT in an SWD, we may derive the **officialOnto** relation from the namespace of an SWT. Swoogle has observed the case that a term t is defined by many ontologies, some of which may not located in the URL mentioned by the URIref of t. For example, *foaf:Person* has been defined as a class by 17 different ontologies as either *rdf:Class* or *owl:Class*. This situation may make it hard for software agent to import the right ontology for a term in the absence of explicit import instruction. Hence we introduce the "official ontology" relation so as to facilitate users (especially software agents)[4] to import the official one. Our experiment shows that only 59% official ontologies could be directly derived by parsing URIref of term. Therefore we collect some heuristics to find the "official ontology" for an SWT T. Table 1 shows experiment results. It is notable that heuristics 2 and 3 help out important ontologies like Dublin Core and FOAF (which is used in many SWDs) even though we only improved the performance to 62.8%.

1. the namespace of T;
2. 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`);
3. 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/`);
4. when none of the above heuristics applies, the "official ontology" does not exist.

[4] The *officialOnto* relation is important for inference engines to load ontologies for external reference automatically

Table 1. Finding official ontologies for 4508 possible namespaces of SWT

Type	number of ns/percent
1. namespace correct	2661(59%)
2. redirected	18(0.4%)
3. single-RDF	150(3.4%)
4. confused	1679(37.2%)

3.3 Relations between Documents

SWDs can be related by properties from meta-ontologies.

- Although not defined explicitly, *rdfs:isDefinedBy* and *rdfs:seeAlso* are widely used in linking to a web document. In practice, many RDF crawlers use *rdfs:seeAlso* to discover SWDs.
- *owl:OntologyProperty* is explicitly defined to associate two ontologies, which are SWDs as well. Swoogle Statistics has discovered much more usage of *owl:imports* than that of the rest instances of *owl:OntologyProperty*. In fact *owl:imports* is also important in showing the dependency between ontologies and is complemented by "officalOnto" relation.

Definition 6. *An SWO* $d1$ **imports** *another* $d1$ *when there is a triple in* $d1$ *in form of (* $d1$*, owl:imports,* $d2$*) or (* $d1$*, daml:imports,* $d2$*).*

Inspired by RDF test-case ontology, we developed a class *wob:RDFDocument*, which explicitly shows the resource is an SWD, to support "hyperlinks" in the semantic web.

4 Ranking Semantic Web Documents

In the semantic web, RDF graphs are usually accessed at document level, and users' navigation essentially jumps from one SWD to another. Therefore, we may simplify semantic web navigation model by generalizing navigation paths into document level. These paths, unlike hyperlinks, lead to non-uniform random surfing behavior.

4.1 Building Document Level Navigation Paths

extension(EX) relation is generalized from the combination of *defines*, *term-extension* and *officialOnto*. An SWD $d1$ *EX* another SWD $d2$ when (i) $d1$ defines a term $t1$, $t1$ extends a term $t2$, and $t2$'s official ontology is $d2$; and (ii) $d1$ and $d2$ are different SWDs.

use-term(TM) relation is generalized from the combination of *uses* and *officialOnto*. An SWD $d1$ *TM-IN* another SWD $d2$ when (i) $d1$ uses a term t, t's official ontology is $d2$; and (ii) $d1$ and $d2$ are different SWDs.

import(IM) relation directly come from *imports* relation.

4.2 Rational Surfer Model and OntoRank

Based on the navigation model, we first developed *OntoRank* with a *rational surfer model*, which emulates user's navigation behavior at document level granularity. The *rational surfer model* inherits random surfer model[3]: a surfer can either navigate from one SWD to another with a constant probability d or jump into an SWD randomly otherwise; but it is also 'rational' since it jumps non-uniformly with the consideration of link semantics. Intuitively, our *OntoRank* estimates the probability of a *rational surfer* will visit an SWD with the bias that ontologies are more preferred to instance data. Let $link(a, l, b)$ be the semantic link from SWD a to SWD b with semantic tag l; $weight(l)$ be user specified navigation preference over semantic links with tag l. We compute *OntoRank* using equation 1 and equation 2.

$$\begin{aligned} wPR(a) &= (1-d) + d \sum_{linkto(x,a)} \frac{wPR(x) \times f(x,a)}{\sum_{link(x,\_,y)} f(x,y)} \\ f(x,a) &= \sum_{link(x,l,a)} weight(l) \end{aligned} \tag{1}$$

Our *rational surfer model* assumes that the surfer, when encounter a SWD D, MUST transitively import the "official" ontologies that defines terms (classes and properties) used by D so as to fully understand D. Hence equation 2 compute the *OntoRank* of a SWD a by summing up the *wPR* of SWDs in $OTC(a)$ (i.e. a set of SWDs that (transitively) import a as ontology).

$$OntoRank(a) = wPR(a) + \sum_{x \in OTC(a)} wPR(x) \tag{2}$$

4.3 OntoRank vs PageRank

In order to evaluate OntoRank, we used a data set collected by Swoogle with 329,987 SWDs including: (i) 4,171 SWOs (ii) 79,784 FOAF documents, and (iii) 196,666 RSS documents. It contains 1,722,412 document level relations including *EX, TM, IM*. Note that these relations are not html 'hyperlinks' but the semantic links derived from triples in SWDs and metadata generated by Swoogle. Then we compose 10 Swoogle document queries by selecting top 10 frequently defined local names (i.e. order by the number of namespaces that define them).

For each local name, we evaluated OntoRank against PageRank by running Swoogle Search to retrieve all relevant SWDs, and then picked up 20 SWDs with highest OntoRank and another 20 SWDs with highest PageRank. Those selected SWDs are classified into two categories, namely ontologies and instance data, based on their ontology ratio[5]. We then counted the number of ontologies(SWOs) in each result set and computed the *difference* using the number of SWOs found by PageRank as reference point.

[5] Ontology Ratio shows the portion of individuals that are defined as classes or properties. For example, given an SWD defining a class "Color' and populating the class with three instances 'blue', 'green' and 'red', its ontology ratio is 25% since only one out of the four individuals is defined as class. High ontology-ratio usually implies the preference of adding term definition rather than populating existing terms. Swoogle use a threshold based heuristics to classify ontologies from other SWDs an ontology's ontology ratio should be no less than 0.8.

The performance of ranking algorithm is evaluated by the number of SWOs in their top 20 ranked SWDs, and experimental result is shown in table 2. It is easy to see that OntoRank outperforms PageRank in finding SWDs with higher ontology-ratio for popular queries. We can observe similar results in other queries as well.

Table 2. OntoRank vs PageRank: OntoRank helps Swoogle Search find more ontologies in top 20 results

query	C1: no. of SWO by OntoRank	C2: no. of SWO 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%

We also compare the overall ranking of SWDs in the data set described above. In table 3, RDFS schema ranks undoubtedly the first according both OntoRank and PageRank. OWL ranks higher than RDF because it is referred by more popular ontologies. RSS and FOAF ontologies rank the 2nd and 4th by PageRank due to their huge amount of instance documents but rank lower by OntoRank due to their limited domain and being less referred by other ontologies. An interesting case is the web of trust (WOT) ontology: it ranks only 29 by PageRank since our data set only contains 280 FOAF documents referring it directly; but it ranks 8th by OntoRank since it is referred by FOAF ontology, which greatly increases WOT's visibility.

5 Ranking for Ontology Dictionary

Ranking terms is a consequential idea after ranking documents and it provides finer granularity. For example, when we query the most popular SWT for "person", *foaf:Person* might be of better interest than another class which is only populated by none or several SWDs. In addition, when describing an instance of *foaf:Person*, *rdfs:seeAlso* is used with certain implicit semantics even though it is not defined in FOAF namespace. These observations lead to the "Do It Yourself" idea for ontology creation i.e. customizing an application oriented ontology by assembling popular terms from multiple ontologies without importing them completely. To this end, we developed an *Ontology Dictionary*, which break ontologies into individual terms. Ontology dictionary addresses similar concerns to the work on splitting ontologies based on logical formalisms [9]; but we focus on higher granularity driven by usage statistics.

Table 3. OntoRank vs PageRank: how top ontologies are ranked

Ontology URL	Ontology Ratio	Onto Rank	Page Rank
http://www.w3.org/2000/01/rdf-schema	94%	1	1
http://purl.org/dc/elements/1.1	100%	2	3
http://www.w3.org/2002/07/owl	86%	3	5
http://www.w3.org/1999/02/22-rdf-syntax-ns	81%	4	6
http://purl.org/rss/1.0/schema.rdf	100%	5	2
http://xmlns.com/foaf/0.1/index.rdf	84%	6	4
http://www.w3.org/2003/01/geo/wgs84_pos	100%	7	10
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

An *application oriented ontology* focuses on encode knowledge using the popular terms to reduce heterogeneity in knowledge sharing. It can be constructed through the following interactions with Swoogle's Ontology Dictionary:

```
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 Terms

Ranking terms facilitates the first step in *CONSTRUCT-ONTO*: when several SWTs are found by *Term Search*, term ranking will help us to find the most useful/popular one.

A straightforward way to evaluate the utility of an SWT is by counting how many times it is used, i.e. the amount of SWDs that *swoogle:uses* it directly (or the amount of their occurrence in all SWDs). Counting SWDs makes an ideal assumption that all SWDs can be uniformly visited; in fact, a more realistic assumption is that SWDs will be visited by their importance, which is approximated by SWD rank. Therefore, we suggest *TermRank* SWTs as shown in equation 3. Intuitively, we split the rank of SWDs to the SWTs populated by them. Given a term t and an SWD d, $TWeight(t,d)$ is computed from $cnt\_uses(d,t)$, which shows how many times d uses t, and $|\{d|uses(d,t)\}|$, which shows how many SWDs use t in our entire SWD collection.

$$\begin{aligned} TermRank(t) &= \sum_{uses(d,t)} \frac{OntoRank(d) \times TWeight(d,t)}{\sum_{uses(d,x)} TWeight(d,x)} \\ TWeight(d,t) &= cnt\_uses(d,t) \times |\{d|uses(d,t)\}| \end{aligned} \tag{3}$$

Table 4 lists top 10 classes having 'person' as local name ordered by TermRank. It is easy to see that *foaf:Person* is significantly highly ranked. An interesting observation is that `http://www.w3.org/2000/10/swap/pim/contact\#Person` ranks higher than `http://xmlns.com/foaf/0.1/person`, which is an error

version of *foaf:Person*, even when the former has been less populated. `http://ebiquity.umbc.edu/v2.1/ontology/person.owl\#Person` could be a possible choice even though it is not populated since the ontology that defines it has significant importance.

Table 4. Top 10 terms about 'person' order by TermRank

Resource URI	#swd_pop	#instance	#swd_def	TermRank
http://xmlns.com/foaf/0.1/Person	74589	1260759	17	979.598257
http://xmlns.com/wordnet/1.6/Person	2658	785133	80	0.88251529
http://www.aktors.org/ontology/portal#Person	267	3517	6	0.00860308
http://www.w3.org/2000/10/swap/pim/contact#Person	257	935	1	0.00714734
http://www.iwi-iuk.org/material/RDF/1.1/Schema/Class/mn#Person	277	398	1	0.00572838
http://xmlns.com/foaf/0.1/person	217	5607	0	0.00259013
http://www.amico.org/vocab#Person	90	90	1	0.00064801
http://www.ontoweb.org/ontology/1#Person	32	522	2	0.00008743
http://ebiquity.umbc.edu/v2.1/ontology/person.owl#Person	0	0	1	0.00005
http://description.org/schema/Person	10	10	0	0.00004479

Table 5 shows the top 10 terms in TermRank order: #swd shows how many SWDs have populated this SWT as specified cat('p' for property, 'c' for class), #instance shows the how many times this SWT is populated. Our $TermRank$ ranks an SWT different from simply counting #swd that populates it, for example, *rdfs:comment* is ranked higher than *dc:title* it is used by many important SWDs and applies to more general context. Another observation is that classes are less popular than properties in the semantic web.

Table 5. Top 10 terms order by TermRank

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

5.2 Ranking RDF Graphs

Since SWD serializes RDF graph, it is natural to extend ranking to the statements asserted by the RDF graph. By tracking the source SWDs that publish a sub-graph, we

may derive its popularity through statistics. One application of such ranking is resolving semantic conflicts. For example, we might have collected two conflicting triples claiming different value for a person's homepage from multiple sources while the cardinality constraint of homepage property is 1; then, we can choose the one with higher ranking.

A more specific problem that directly relates to step 2 in *CONSTRUCT-ONTO* is ranking *class-property bonds* .

Definition 7. *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: $(\_x, rdf : type, class), (\_x, property, \_)$.

Ranking c-p bonds help users to choose the most popular properties for a class when they are publishing data with the desire of maximizing the data's 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*. To rank c-p bonds, we cannot rely on the definition from ontologies, which does not show how well a bond is adopted; instead, we need to look at the existing instances of *foaf:Person* and summarize their usage for reference. Table 6 shows some commonly used properties of *foaf:Person*.

Table 6. Top 5 properties for foaf:Person

property	sources
foaf:mbox_sha1sum	67136
foaf:nick	62266
foaf:weblog	54341
rdfs:seeAlso	47228
foaf:name	46590

6 Related Work

In literature, there are two widely used approaches to ranking web documents: (i) *content analysis* ranks documents using various content models such as vector-space model [10]; (ii) *link analysis* ranks web documents using various graph navigation models, such as PageRank [3, 11] and HITS [12, 13]. Unfortunately, they are not aware of the "semantic markup" in SWD or "semantic link" between SWDs.

Ranking terms is a special problem brought by the semantic web, where many URIrefs may refer to the same concept. The closest works in literature is ranking popular namespace or terms by counting their references [14, 15].

Ranking triples has two well known approaches: (i) *content analysis* ranks triples by combining weight to different SWTs according to users specified interests [16] or outliers-discovery heuristics [17]; (ii) *context analysis* ranks triples using provenance information. It falls in semantic web trust layer research [18].

7 Discussion and Evaluation

Swoogle has been running as a web-based service since the Spring of 2004. Swoogle is intended to support two use cases: (1) supporting human "knowledge engineers" with services to help find appropriate ontologies and knowledge and to understand how these are being used on the semantic web and (2) providing software agents and tools with services to find knowledge and data on the semantic web. While we have not yet done any formal evaluations of Swoogle, we offer some observations and comments that address the questions of how well Swoogle meets its goals and informally compare it to the alternatives.

First, we point out that Swoogle is an operational system with a large number of people who report to us that they use it regularly to support their work on developing semantic web based systems. We estimate that this "customer base" consists of hundreds of regular users and thousands of more casual ones. The feedback we've received has been very positive.

Swoogle's database currently has information on about 340,000 semantic web documents which contain almost 48 million triples and define approximately 97,000 classes, 54,000 properties and 7,000,000 individuals. Just over 4,000 of these documents are 'ontologies' that mostly define classes and properties as opposed to mostly asserting facts about individuals. Currently, the most popular kinds of documents are FOAF files and RSS files. We have discovered many more SWDs, most of which are simple FOAF or RSS documents, and have chosen not to add these to Swoogle's current database in order to keep the dataset interesting and balanced. A new version of Swoogle is under development that will include significantly more data.

There are three kinds of alternatives to Swoogle that can be used to find knowledge on the semantic web: using conventional search engines such as Google, and using specialized portals and repositories such as semwebcentral.org and schemaweb.org, and accessing specialized collections such as several for FOAF and RSS documents.

Some conventional search engines do index RDF documents and some do not. Those that do, including Google, can be used to find SWDs and SWTs. However, none of them understand the content they are indexing, recognize that some of the terms are links to other documents, or even correctly parse RDF documents in any of the standard encodings (e.g., XML, N3, Turtle). Any ranking that is done by such systems completely ignores links between SWDs and the corresponding semantic relationships.

There are some useful web-based repositories available for semantic web documents. All of these, to our knowledge, require that people manually suggest documents to be added and provide appropriate meta-data. Thus they tend to be small and have poor coverage, although the quality of the submitted documents is high.

Finally, there are several crawler-based systems which are specialized to particular kinds of RDF documents – namely FOAF documents and RSS documents. These are narrow in their scope and do not have the same goals and services in mind.

A formal evaluation of how well Swoogle performs on finding and ranking SWDs and SWTs would be based, in part, on measuring it's precision and recall using standard techniques developed for information retrieval systems. This would allow us to compare Swoogle's performance to using other systems (e.g., Google) and also to compare variations on our ranking algorithm and to evaluate the importance and effectiveness of

doing more or less inference when analyzing SWDs and SWTs. While we intend to carry out such an evaluation in the future, it will require careful design and significant labor to acquire the necessary human evaluations as a baseline.

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: Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management. (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. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
4. Haveliwala, T.H.: Topic-sensitive pagerank. In: WWW '02: Proceedings of the eleventh international conference on World Wide Web, ACM Press (2002) 517–526
5. Jeh, G., Widom, J.: Scaling personalized web search. In: WWW '03: Proceedings of the twelfth international conference on World Wide Web, ACM Press (2003) 271–279
6. H.Zhuge, Zheng, P.: Ranking semantic-linked network. In: www 2003. (2003)
7. BaezaYates, R., Davis, E.: Web page ranking using link attributes. In: www 2004. (2004)
8. Patel, C., Supekar, K., Lee, Y., Park, E.K.: Ontokhoj: a semantic web portal for ontology searching, ranking and classification. In: WIDM '03: Proceedings of the 5th ACM international workshop on Web information and data management, ACM Press (2003) 58–61
9. Grau, B.C., Parsia, B., Sirin, E.: Working with multiple ontologies on the semantic web. In: Proceedings of the Third Internatonal Semantic Web Conference (ISWC2004). Volume 3298 Lecture Notes in Computer Science. (2004)
10. Salton, G., McGill, M.J.: An Introduction to Modern Information Retrieval. McGraw-Hill (1983)
11. Haveliwala, T.: Efficient computation of pageRank. Technical Report 1999-31, Stanford University (1999)
12. Kleinberg, J.: Authoritative sources in a hyperlinked environment. In: Proceedings of ACM-SIAM Symposium on Discrete Algorithms. (1998)
13. Chakrabarti, S., Dom, B.E., Kumar, S.R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., Kleinberg, J.: Mining the web's link structure. Computer **32** (1999) 60–67
14. Eberhart, A.: Survey of rdf data on the web. Technical report, International University in Germany (2002)
15. Ding, L., Zhou, L., Finin, T., Joshi, A.: How the semantic web is being used:an analysis of foaf. In: Proceedings of the 38th International Conference on System Sciences. (2005)
16. Aleman-Meza, B., Halaschek, C., Arpinar, I.B., Sheth, A.: Context-aware semantic association ranking. In: First International Workshop on Semantic Web and Databases. (2003)
17. Anyanwu, I.K., Maduko, A., Sheth, A.: Semrank: Ranking complex relationship search results on the semantic web. In: WWW '05: The 14th International World Wide Web Conference. (2005) to appear.
18. Carroll, J., Bizer, C.: The semantic web trust layer. http://www.wiwiss.fu-berlin.de/suhl/bizer/pub/carrollbizer-trust-www2004-devday.pdf (2004)