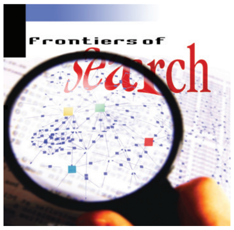
Search on the Semantic Web



To help human users and software agents find relevant knowledge on the Semantic Web, the Swoogle search engine discovers, indexes, and analyzes the ontologies and facts that are encoded in Semantic Web documents.

Li Ding
Tim Finin
Anupam Joshi
Yun Peng
Rong Pan
Pavan Reddivari
University of Maryland, Baltimore County

Search engines have assumed a central role in the World Wide Web's infrastructure as its scale and impact have increased. In the Web's earliest days, people found pages of interest by navigating (quickly dubbed *surfing*) from pages whose locations they remembered or bookmarked. Rapid growth in the number of pages gave rise to Web directories like Yahoo that manually organized Web pages into a hierarchy of topics.

As the growth continued, these directories were augmented by search engines such as Lycos, HotBot, and AltaVista, which automatically discovered new and modified Web pages, added them to databases and indexed them by their keywords and features. Today, search engines such as Google and Yahoo dominate the Web's infrastructure and largely define our Web experience.

Most knowledge on the Web is presented as natural-language text with occasional pictures and graphics. This is convenient for human users to read and view but difficult for computers to understand. It also limits the indexing capabilities of state-of-the-art search engines, since they cannot infer meaning—for example, does an occurrence of the word "raven" refer to the bird or to Baltimore's football team?

Thus, users share a significant burden in terms of constructing search queries intelligently. Even with increased use of XML-encoded information, computers still must use application-dependent semantics to process the tags and literal symbols.

SEMANTIC WEB SEARCH

The Semantic Web offers an approach in which computers can use symbols with well-defined, machine-interpretable semantics to share knowledge.[1] Search on the Semantic Web differs from conventional Web search for several reasons.

First, Semantic Web knowledge content is intended for publication by machines for machines—tools, Web services, software agents, information systems, and so forth. Although Semantic Web annotations and markup can help users find human-readable documents, there will likely be an "agent layer" between human users and Semantic Web search engines.

Second, knowledge encoded in Semantic Web languages such as the Resource Description Framework (RDF)[2] differs from both the largely unstructured free text found on most Web pages and the highly structured information found in databases. Such semistructured information requires using a combination of techniques for effective indexing and retrieval. RDF, RDF Schema (RDFS),[3] and the Web Ontology Language (OWL)[4] introduce semantic features beyond those used in ordinary XML, allowing users to define terms (for example, classes and properties), express relationships among them, and assert constraints and axioms that hold for well-formed data.

Third, even within a single document, Semantic Web documents (SWDs) can be a mixture of concrete facts, class and property definitions, logic constraints, and metadata. Fully understanding the

Published by the IEEE Computer Society

document can require substantial reasoning, so developers must face the design issue of how much reasoning search engines can do and when they should do it. This reasoning produces additional facts, constraints, and metadata that may also need to be indexed, potentially along with the supporting justifications. Conventional search engines do not try to understand document content because the task is just too difficult and requires more research on text understanding.

Finally, the graph structure of a collection of online SWDs differs significantly from the structure that emerges from a collection of HTML documents. This difference influences both the development of effective strategies for automatically discovering online Semantic Web documents and the establishment of appropriate metrics for ranking their importance.

Rather than using one uniform crawling technique to discover SWDs, Swoogle, a Semantic Web search engine developed by the eBiquity group at UMBC,[5] employs a fourfold strategy:

- running metasearches on conventional Web search engines, such as Google, to find candidates;
- using a focused Web crawler to traverse directories in which SWDs have been found;
- harvesting URLs when processing discovered SWDs; and
- collecting URLs of SWDs and directories containing SWDs that users have submitted.

To help human users and software agents find relevant knowledge on the Semantic Web, Swoogle discovers, indexes, and analyzes the ontologies and facts that are encoded in SWDs.

THE SEMANTIC WEB FRAMEWORK

The Semantic Web is a framework that allows computers to publish, share, and reuse data and knowledge on the Web and across application, enterprise, and community boundaries. It is a collaborative effort led by the World Wide Web Consortium based on the layered set of standards. Figure 1 shows a simple Semantic Web document encoded using the RDF/XML syntax.[6]

```
 1:  <?xmlversion="1.0"encoding="utf-8"?>
 2:  <rdf:RDFxmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 3:           xmlns:owl="http://www.w3.org/2002/07/owl#"
 4:           xmlns:foaf="http://xmlns.com/foaf/0.1/">
 5:     <foaf:Person>
 6:        <foaf:name>LiDing</foaf:name>
 7:        <foaf:mbox rdf:resource="mailto:dingli1@umbc.edu"/>
 8:        <owl:sameAsrdf:resource="http://www.csee.umbc.edu/~dingli1/foaf.rdf#dingli"/>
 9:     </foaf:Person>
10:  </rdf:RDF>
```

Figure 1. An example Semantic Web document written in RDF/XML. The document is available at http://ebiquity.umbc.edu/get/a/resource/134.rdf.

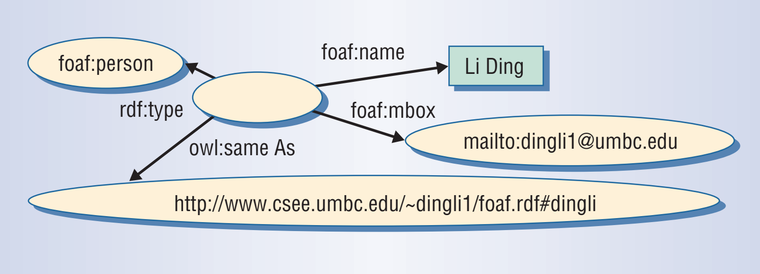


Figure 2. The RDF graph of the foaf:Person instance.

Line 1 declares that this is an XML document. Lines 2-4 further define the content to be an RDF document and provide abbreviations for three common "namespaces" for RDF, OWL, and Friend of a Friend (FOAF), which define classes and properties for describing people, their common attributes, and relations among them. The SWD's vocabulary consists of literals ("Li Ding" in line 6), URI-based resources (mailto:dingli1@umbc.edu in line 7), and anonymous resources (lines 5-9). Users assert statements using RDF triples such as the one in line 6, which asserts of the anonymous foaf:Person subject introduced by Line 5 that it has a foaf:name property with the value "LiDing."

A higher level of granularity is class-instance, which is supported by the object-oriented ontology constructs in RDFS. Figure 2 is an RDF graph stating that there is an instance of a foaf:Person having foaf:name "Li Ding," foaf:mbox mailto: dingli1@umbc.edu, and this instance is owl:sameAs another class-instance identified by http://www.csee.umbc.edu/~dingli1/foaf.rdf#dingli.

The Semantic Web can be thought of as a collection of loosely federated databases that separates physical Web storage (realized by online SWDs) from the logical representation (conveyed by the RDF graph model). In this view, the Semantic Web represents a large, universal RDF graph whose parts are physically serialized by SWDs distributed across the Web. However, the formal semantics associated with Semantic Web languages support generating new facts from existing ones, while conventional databases only enumerate all facts.

SEARCH ENGINE TASKS

Search engines for both the conventional Web and the Semantic Web involve the same set of high-level tasks: discovering and revisiting online documents, processing users' queries, and ordering search results. Their details diverge, however, due to differences in the distribution of SWDs and the semantics of their content.

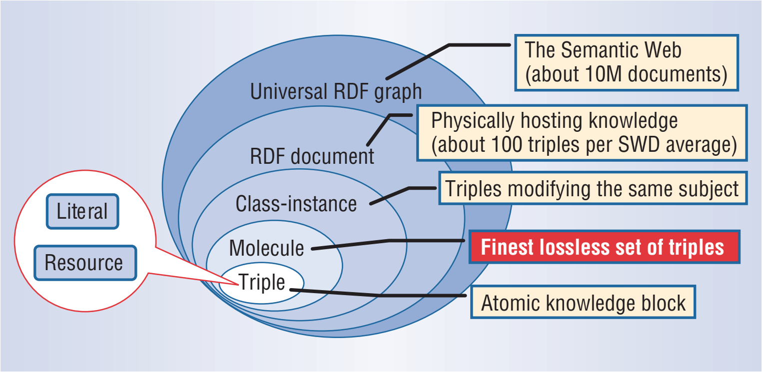


Figure 3. Semantic Web granularity. The granularity levels range from the universal graph comprising all RDF data on the Web to individual triples and their constituent resources and literals.

Discovering and revisiting documents

Conventional search engines either scan all possible IP addresses or employ crawlers to discover new Web documents. A typical crawler starts from a set of seed URLs, visits documents, and traverses the Web by following the hyperlinks found in the visited documents. The fact that the Web forms a well-connected graph and that people can manually submit new URLs make this an effective process.

A Semantic Web crawler must deal with several problems. SWDs are needles in the Web's haystack, so an exhaustive crawl of the Web is not an efficient approach. Moreover, an SWD graph is not yet as dense and well-connected as a conventional Web page graph, so starting with a few seeds is unlikely to yield many SWDs.

One approach is for a Semantic Web crawler to use conventional search engines to discover a large number of potential seed SWDs. These then need to be validated by a semantic parser. Finally, many of the URLs found in an SWD point to documents that are not SWDs, so heuristics to limit and prune candidate links are beneficial.

For the most part, the issue of how often to revisit documents to monitor changes is the same for both the conventional Web and the Semantic Web. However, modifying an SWD can have far-reaching effects if class or property definitions used by other documents are changed. Depending on the nature and amount of reasoning done when analyzing and indexing documents, updating an SWD can trigger significant work for a Semantic Web search engine.

Query processing

A search engine's core task is processing queries against the data it has indexed. This can be broken down into three issues: What should be returned as query results? Over what data should queries be run? What constraints can be used in a query? As Figure 3 shows, the Semantic Web can aggregate data at several levels of granularity, ranging from the universal graph of all RDF data on the Web to a single RDF triple and the term URIs it comprises. Unlike the Web, where users usually search knowledge at the document granularity, the Semantic Web can be queried at various levels of granularity, including the following:

- *RDF database search*. At the unified RDF graph level, the Semantic Web is essentially a simple database of triples, and search is done by processing semistructured query languages like RDQL and SPARQL.
- *Semantic Web document search*. We often want to query the Semantic Web to find relevant SWDs. This helps users filter out huge amounts of irrelevant Semantic Web knowledge and promotes the emergence of consensus ontologies, which define common terms for sharing and reusing knowledge. In comparison with Web search, the query results are also document URLs, but the query constraints are not simply keywords.
- *RDF subgraph search*. While RDF triples are physically grouped by SWDs, they can be also logically grouped by named graph,[7] resource description (that is, a collection of triples with a common subject), or an RDF molecule.[8] Search at this level is a refinement of the above two searches.
- *Semantic Web vocabulary search*. At the URI level, Semantic Web vocabulary terms (that is, URIrefs) are analogous to words in natural language. Like dictionary lookup, searching appropriate terms for a concept is critical to query composition in all the above searches.

The metadata for a Semantic Web document should include metadata about itself (such as document URL and last-modified time) and its content (such as terms being defined or populated and ontology documents being imported).

Ranking

Google was the first search engine to order its search results based in part on a Web page's "popularity" as computed from the Web's graph structure. This idea has turned out to be enormously useful in practice and is equally applicable to Semantic Web search engines.

However, Google's PageRank[9] algorithm for ranking Web pages cannot be directly used in the Semantic Web for several reasons. While all HTML links are essentially the same, Semantic Web links come in many varieties, each with semantics that may affect a ranking algorithm. Moreover, conventional search engines only rank Web pages, whereas a Semantic Web search engine must rank SWDs as well as RDF graphs, triples, and terms. Each admits different ranking models and algorithms.

SWOOGLE SEMANTIC WEB DISCOVERY

Swoogle currently uses Google to find a large number of initial "seed" documents that are likely to be SWDs. Other seeds come from user submissions. Since SWDs typically use special file extensions such as .rdf or .owl, Swoogle queries for files with such extensions. The extensions are dynamically selected (an extension is selected if more than 10 SWDs have used it and it has at least 50 percent accuracy in classifying SWDs).

Since Google returns at most 1,000 results for any query, Swoogle takes advantage of its feature that restricts a search to results from a specified domain or site. Site queries work because of the locality hypothesis—a Web site hosting more than two SWDs is likely to have more.

Swoogle uses the Jena2 parser (www.hpl.hp.com/semweb/jena2.htm) to validate that the files Google returns are SWDs. Once it has discovered an SWD, Swoogle uses a simple focused crawler to explore the Web environment around it to find more. After filtering out the non-SWDs from the results, Swoogle extracts a list of the sites on which the SWDs were found and uses them as seeds for further crawls as well.

Discovery results

In 2002, Andreas Eberhard reported 1,479 SWDs with about 255,000 triples out of nearly 3×10^6 Web pages.[10] As of July 2005, Swoogle had found more than 5×10^5 SWDs with more than 7×10^7 triples. Although this number is far less than Google's 8×10^9 Web pages, it represents a nontrivial collection of Semantic Web data.[11]

Figure 4 plots a power law distribution of last-modified time of SWDs (swd curve), which demonstrates that the Semantic Web is either experiencing a rapid growth rate or, at the very least, is being actively maintained.

The apparent growth in the number of ontology documents (onto curve) is somewhat biased by the SWDs using the Inference Web namespaces (for example, http://inferenceweb.stanford.edu/2004/07/iw.owl), which are intended to be instance documents but, unfortunately, include many unnecessary class/property definitions. After removing such documents, the distribution (onto* curve) ends with a much flatter tail. This in part indicates a trend away from ontology development to populating and reusing ontologies.

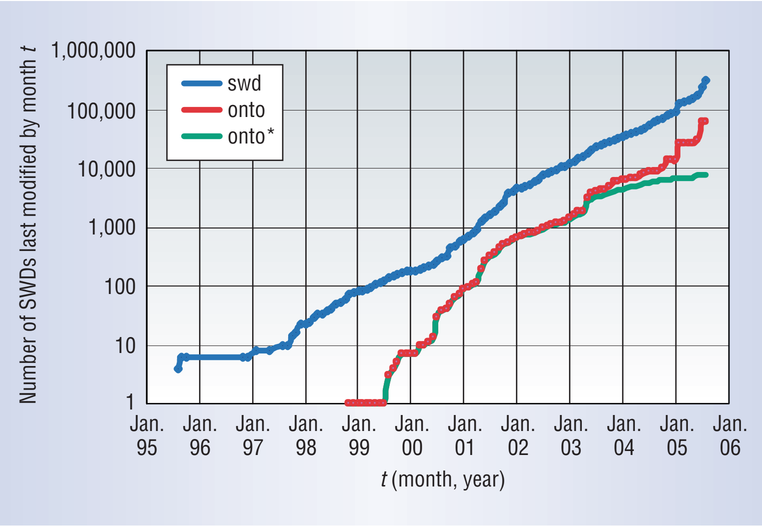


Figure 4. The number of SWDs and ontologies last modified by month t.

SWOOGLE SEMANTIC WEB SEARCH

Swoogle concentrates on Semantic Web document and vocabulary searches, which emphasize the Web aspects of the Semantic Web. It differs from an RDF database or RDF subgraph search in that it maintains compact metadata about documents and terms without recording all encountered triples. In addition to the conventional metadata obtained without semantic parsing, Swoogle indexes the semantic content of and relations among SWDs and terms.

To find SWDs, Swoogle supports constraints on the following metadata:

- *Document level metadata*. For example, users can search for SWDs using .rdf as the file extension.
- *Semantic content metadata*. For example, users can search for SWDs using RDF/XML as the syntax language. Users can also search for SWDs intended to be an ontology. This is supported by an SWD's *ontology ratio*—that is, the fraction of its class-instances being recognized as classes and properties. Swoogle considers an SWD to be an ontology document (SWO) if its ontology ratio exceeds an empirical threshold (that is, 0.8);
- *Relational metadata*. For example, users can search for SWDs that use the FOAF namespace or that define instances of the foaf:Person class.

To find Semantic Web vocabulary terms, Swoogle supports two types of queries:

- *Search for terms*. For example, the terms for the concept "actor" can be found by checking

Table 1. Comparison of OntoRank and PageRank in finding ontology documents.

Term	Ontologies found by OntoRank (C1)	Ontologies found by PageRank (C2)	Difference: percent (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**

if the local-names of their URIref have the word "actor" but not "factor."
- *Search for resource description.* When looking for a set of attributes that modifies foaf:Person, users can search Swoogle's ontology dictionary for properties having the class foaf:Person as their domain. Such domain relations can be obtained by parsing relevant ontology documents as well as reverse-engineering the class-instances of foaf:Person.

To our surprise, reverse engineering the foaf:Person class-instances resulted in more than 500 properties, whereas parsing relevant ontology documents returned only 167 findings.

SWOOGLE SEMANTIC WEB RANKING

Google's success with its PageRank algorithm has demonstrated the importance of ordering the results that a query returns. Swoogle uses two custom ranking algorithms—OntoRank and TermRank—to order a collection of SWDs or terms, respectively. These algorithms are based on an abstract "surfing" model that captures how an agent might access Semantic Web knowledge. Navigational paths on the Semantic Web are defined by RDF triples as well as by the resource-SWD and SWD-SWD relations. However, revealing most of these connections requires a centralized analysis.

Ranking SWDs using OntoRank

Since a Web document is the primary unit of data access on the Web, Swoogle aggregates navigational paths to the SWD level and recognizes three generalized interdocument links.[12]

- An extension relation holds between two SWDs when one defines a term by using terms defined in another.
- A use-term relation holds between two SWDs when one uses a term that another defines.
- An import relation holds when one SWD imports, directly or transitively, another SWD.

Google's simple *random surfer model* is not appropriate for these paths. For example, an agent reasoning over the content found in an SWD should access and process all of the ontologies it imports. 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 jumps to a new random SWD with a constant probability or follows a link in the current SWD to another SWD. However, it is "rational" in that it follows a link nonuniformly and in accord with link semantics: When encountering an SWD, the rational surfer will (transitively) import the "official" ontologies that define the classes and properties the SWD references.

We used a data set containing 330,000 SWDs (1.5 percent SWOs, 24 percent FOAF documents, and 60 percent RSS documents) and 200,000 document-level relations to compare the effectiveness of PageRank and OntoRank in finding ontologies. Ten popular local-names (according to Swoogle's statistics) were selected as queries. For each query, we sorted the results using PageRank and OntoRank individually and then compared the number of ontology documents among the top 20 highest ranked results by both PageRank and OntoRank. Table 1 shows that OntoRank outperformed PageRank by an average of 40 percent.

Ranking terms

Swoogle uses TermRank to sort terms by their popularity. This can be measured by the number of SWDs using or populating a term. This approach, however, ignores agents' rational behavior in accessing SWDs, so the SWDs' OntoRank values modulate the results. Table 2 lists TermRank's ordering of the 10 highest ranked classes having "person" as the local-name. Not surprisingly, the foaf:Person class is number one. The sixth term is a common mistyping of the first one, so it has been populated without being defined. The ninth term appears in the list by virtue of the high OntoRank score of the ontology that defines it.

APPLICATIONS

To explore what services a Semantic Web search engine can provide, we have used Swoogle to support several applications and use cases. These projects include helping researchers find ontologies and data, semantic search over documents representing proofs, and finding and evaluating semantic asso-

Table 2. Top ten results when searching for classes with "person" in their local-name.

Rank	Resource URI	No. of SWDs populating term	No. of instances	No. of SWDs defining term
1	http://xmlns.com/foaf/0.1/Person	74,589	1,260,759	17
2	http://xmlns.com/wordnet/1.6/Person	2,658	785,133	80
3	http://www.aktors.org/ontology/portal#Person	267	3,517	6
4	ns1:Person[1]	257	935	1
5	ns2:Person[2]	277	398	1
6	http://xmlns.com/foaf/0.1/Person	217	5,607	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] http://www.w3.org/2000/10/swap/pim/contact#

[2] http://www.iwi-iuk.org/material/RDF/1.1/Schema/Class/mn#

[3] http://ebiquity.umbc.edu/v2.1/ontology/person.owl#

ciations in large graph databases.

In the NSF-supported SPIRE project, a group of biologists and ecologists is exploring how to use the Semantic Web to publish, discover, and reuse models, data, and services.[13] Researchers need to find appropriate ontologies and terms for annotating their data, and they also need resources for discovering data and services others have published.

With Swoogle's ontology search interface, users can search for existing ontology documents that define terms in which user-supplied keywords are the substring of their local-name. For example, to find an ontology for describing temporal relations, the search might use the keywords "before," "after," and "interval." Swoogle's ontology dictionary provides definitions for a given property (or class). It can assemble and merge definitions from multiple sources, list terms sharing the same namespace or the same local-name, and list domain associations between classes and properties. Those associations can either be "ontological" (for example, the foaf:knows property is defined as existing between instances of foaf:person), or "empirical" (for example, applying the dc:creator property to an instance of foaf:Person). Judging the ranking or popularity of terms and ontologies is also relevant. Community consensus models as reflected in ontologies tend to be ranked highly, thus searches use them more often.

Researchers are using Swoogle in conjunction with the Inference Web (IW),[14] which explicitly represents proofs using Proof Markup Language (PML),[15] an OWL ontology. One IW component, IWSearch (http://iw4.stanford.edu/iwsearch/IWSearch), uses Swoogle document search (searching SWDs using IW namespaces) to discover newly published or updated PML documents on the Web and itself is powered by a specialized instance of Swoogle to index and search instances found in a corpus of more than 50,000 PML documents. Indexing the conclusion part of a proof NodeSet instance can lead to the discovery of additional NodeSets sharing the same conclusion as the one from the given justification tree, thus helping to expand the justification tree with additional proofs.

SEMDIS, an NSF project jointly conducted with researchers at the University of Georgia is also using Swoogle. This project is automating the discovery, merging, and evaluation of complex semantic associations in RDF data drawn from a variety of information sources. SEMDIS augments information collected from the Semantic Web with additional data extracted from text documents and databases.[16] The result, encoded as a large RDF graph along with provenance assertions and trust information, is processed to discover and evaluate "interesting" semantic associations.[17] SEMDIS conducts two kinds of Semantic Web searches:

- searching for a semantic association (connected subgraph) in the large-scale RDF graph, and
- searching SWDs that (partially) support a given semantic association.

The first kind of search finds paths between two nodes in a graph, a common issue in RDF databases. The second is a provenance search to find a set of SWDs that (partially) imply a hypothesized semantic association. Researchers have prototyped this type of search as an RDF molecule-based approach at the RDF subgraph search level.[8]

As the Web has grown in size, search engines have become a critical component of its infrastructure, and there is an increasing need for search engines that can efficiently handle Semantic Web content. While we cannot be sure what form this content will take in the future, the current standard is based on Semantic Web documents. We continue to use Swoogle to study the growth and characteristics of the Semantic Web and the use of RDF and OWL. We are also developing new features and capabilities and exploring how it can be used in novel applications. Many open issues remain.

One set of open problems involves scale. Techniques that work today with 5×10^6 documents may fail when the Semantic Web has 5×10^8 documents. Extending Swoogle to index and effectively query large amounts of instance data remains a challenge. We estimate that the SWDs currently on the Web contain more than 5×10^8 triples, a number that neither current relational databases nor custom triple stores can handle efficiently.

Some of these problems could potentially be solved by moving away from the conventional database technology we are using and creating custom-designed index stores and distributed systems—analogous to what Google has done for conventional Web searches. It remains to be seen, however, if that alone would suffice. We are also interested in developing a query system that can be used to find RDF molecules in a reasonably efficient manner.[8]

We also need to explore how much and where a Semantic Web search engine should reason over the contents of documents and queries. In an earlier system,[18] we experimented with expanding documents using reasoning prior to indexing. A complementary approach is to expand queries containing RDF terms.[19] This is related in part to the problem of scale—the larger the collection becomes, the less efficient it is to reason over it.

Other issues involve trust and the use of local knowledge that is not part of the Semantic Web. Information encoded in RDF is now being embedded in other documents, such as PDF and XHTML documents, JPEG images, and Excel spreadsheets. When techniques for such embedding become standard, we expect the growth of Semantic Web content on the Web to accelerate dramatically. This will add a new requirement for hybrid information retrieval systems that can index documents based on words as well as RDF content. More information about these issues, as well as Swoogle, can be found in a companion technical report.[20] ■

References

1. T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, May 2001, pp. 35-43.
2. G. Klyne and J.J. Carroll, *Resource Description Framework (RDF): Concepts and Abstract Syntax,* Feb. 2004; http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.
3. D. Brickley and R.V. Guha, *RDF Vocabulary Description Language 1.0: RDF Schema*, Feb. 2004; http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.
4. M. Dean and G. Schreiber, *OWL Web Ontology Language Reference*, Feb. 2004; http://www.w3.org/TR/2004/REC-owl-ref-20040210/.
5. L. Ding et al., "Swoogle: A Search and Metadata Engine for the Semantic Web," *Proc. 13th ACM Conf. Information and Knowledge Management*, ACM Press, 2004, pp. 652-659.
6. D. Beckett, *RDF/XML Syntax Specification (Revised)*, Feb. 2004; http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/.
7. J.J. Carroll et al., *Named Graphs, Provenance, and Trust*, tech. report HPL-2004-57, HP Lab, May 2004.
8. L. Ding et al., *Tracking RDF Graph Provenance Using RDF Molecules*, tech. report TR-CS-05-06, UMBC, Apr. 2005.
9. L. Page et al., *The PageRank Citation Ranking: Bringing Order to the Web*, tech. report, Stanford Digital Library Technologies Project, 1998.
10. A. Eberhart, *Survey of RDF Data on the Web*, tech. report, Int'l Univ. in Germany, 2002.
11. Y. Guo, Z. Pan, and J. Heflin, "An Evaluation of Knowledge Base Systems for Large OWL Datasets," *Proc. Int'l Semantic Web Conf.*, Springer, 2004, pp. 274-288.
12. L. Ding et al., "Finding and Ranking Knowledge on the Semantic Web," *Proc. Int'l Semantic Web Conf.* (ISWC), Springer, 2005, to appear.
13. T. Finin and J. Sachs, "Will the Semantic Web Change Science?" *Science Next Wave*, Sept. 2004; http://nextwave.sciencemag.org.
14. P. Pinheiro da Silva, D.L. McGuinness, and R. McCool, "Knowledge Provenance Infrastructure," *Data Eng. Bull.*, vol. 26, no. 4, 2003, pp. 26-32.
15. P. Pinheiro da Silva, D.L. McGuinness, and R. Fikes, *A Proof Markup Language for Semantic Web Services*, tech. report KSL04-01, Stanford Univ., 2004.
16. B. Aleman-Meza et al., "Context-Aware Semantic Association Ranking," *Proc. 1st Int'l Workshop Semantic Web and Databases*, 2003, pp. 33-50.
17. A. Sheth et al., "Semantic Association Identification and Knowledge Discovery for National Security Applications," *J. Database Management*, vol. 16, no. 1, 2005, pp. 33-53.
18. T. Finin et al., "Information Retrieval and the Semantic Web," *Proc. 38th Int'l Conf. System Sciences*, IEEE CS Press, 2005, p. 113.
19. E.M. Voorhees, "Query Expansion Using Lexical-Semantic Relations," *Proc. 17th Int'l Conf. Research*

and Development in Information Retrieval (SIGIR 94), Springer-Verlag, 1994, pp. 61-69.
20. L. Ding et al., *Swoogle Semantic Web Search*, tech. report TR-CS-05-09, UMBC, Sept. 2005.

Li Ding *is a PhD student in computer science and electrical engineering at the University of Maryland, Baltimore County. His research interests include the Semantic Web, intelligent agents, and data mining. He is a member of the American Association of Artificial Intelligence. Contact him at ding.li@umbc.edu.*

Tim Finin *is a professor of computer science and electrical engineering at the University of Maryland, Baltimore County. His research interests include applications of artificial intelligence to problems in information systems, software agents, the Semantic Web, and mobile computing. He received a PhD in computer science from the University of Illinois, Urbana-Champaign. He is a member of the American Association of Artificial Intelligence and the ACM. Contact him at finin@umbc.edu.*

Anupam Joshi *is a professor of computer science and electrical engineering at the University of Maryland, Baltimore County. His research interests include mobile and pervasive computing, the Semantic Web, and security. He received a PhD in computer science from Purdue University. He is a member of the IEEE, the IEEE Computer Society, and the ACM. Contact him at joshi@umbc.edu.*

Rong Pan *is a PhD student in computer science and electrical engineering at the University of Maryland, Baltimore County. His research interests include probabilistic reasoning, the Semantic Web, and information retrieval. He is a member of the American Association of Artificial Intelligence. Contact him at pan.rong@umbc.edu.*

Yun Peng *is an associate professor of computer science and electrical engineering at the University of Maryland, Baltimore County. His research interests include the representation of and reasoning with uncertainty in intelligent systems, machine learning, and the Semantic Web. He received a PhD in computer science from the University of Maryland, College Park. He is a member of the American Association of Artificial Intelligence and the International Neural Network Society. Contact him at ypeng@umbc.edu.*

Pavan Reddivari *is an MS student in computer science and electrical engineering at the University of Maryland, Baltimore County. His research interests include the Semantic Web, pervasive computing, and intelligent agents. He is a member of the American Association of Artificial Intelligence. Contact him at pavan2@umbc.edu.*