Information retrieval on the Semantic Web: Integrating inference and retrieval

James Mayfield
The Johns Hopkins University
Applied Physics Laboratory
Laurel MD 20723-6099 USA
james.mayfield@jhuapl.edu

Tim Finin
University of Maryland,
Baltimore County
Baltimore MD 21250 USA
finin@umbc.edu

ABSTRACT
One vision of the Semantic Web is that it will be much like the Web we know today, except that documents will be enriched by annotations in machine understandable markup. These annotations will provide metadata about the documents as well as machine interpretable statements capturing some of the meaning of document content. We discuss how the information retrieval paradigm might be recast in such an environment. We suggest that retrieval can be tightly bound to inference. Doing so makes today's Web search engines useful to Semantic Web inference engines, and causes improvements in either retrieval or inference to lead directly to improvements in the other.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Retrieval models

General Terms
Management, Measurement, Documentation, Performance, Design, Experimentation, Languages.

Keywords
Semantic Web, information retrieval, DAML+OIL, OWL

1. INTRODUCTION

The Semantic Web [5] has lived its infancy as a clearly delineated body of Web documents. That is, by and large researchers working on aspects of the Semantic Web knew where the appropriate ontologies resided and tracked them using explicit URLs. When the desired Semantic Web document was not at hand, one was more likely to use a telephone to find it than a search engine. This closed world assumption was natural when a handful of researchers were developing DAML 0.5 ontologies, but is untenable if the Semantic Web is to live up to its name.

This work was partially supported by DARPA contract F30602-97-1-0215.

Yet simple support for search over Semantic Web documents, while valuable, represents only a small piece of the benefits that will accrue if search and inference are considered together. We believe that Semantic Web inference can improve traditional text search, and that text search can be used to facilitate or augment Semantic Web inference. Several difficulties, listed below, stand in the way of this vision.

Current Web search techniques are not directly suited to indexing and retrieval of semantic markup. Most search engines use words or word variants as indexing terms. When a document written using some flavor of SGML is indexed, the markup is typically ignored. Because the Semantic Web is expressed entirely as markup, it is invisible to the major Web search engines. Nonetheless, while it is possible that special purpose Web retrieval engines will arise that focus on retrieval of Semantic Web pages, it seems unlikely that they will overtake the coverage of the text search engines in the near future. Thus, we would like to find a way to exploit the capabilities of today's text-based search engines for use with semantic markup.

Current Web search techniques cannot use semantic markup to improve text retrieval. Web search engines typically rely on simple term statistics to identify which documents are most relevant to a query. One might consider techniques such as thesaurus expansion or blind relevance feedback to be integration of inference into the retrieval process, but such inference is simple compared with what is possible using semantic markup. One would like the presence of semantic markup in either the query or the documents retrieved to be exploitable during search to improve that search.

Likewise, text is not useful during inference. To the extent that it is possible to automatically convert text to a semantic representation, such resulting representations can be used during inference. However, semantic interpretation is difficult at best, and unsolved in the general case. We would like a way to exploit relevant text during inference, without needing to analyze the semantics of that text.

There is no current standard for creating or manipulating documents that contain both HTML text and semantic markup. There are two prime candidates for such hybrid documents. First, semantic markup might be embedded directly in an HTML page. Unfortunately, while we call approaches like DAML+OIL and OWL semantic *markup*, they are typically used not as markup but rather as stand-alone knowledge representation languages that are not directly tied to text. Furthermore, embed-

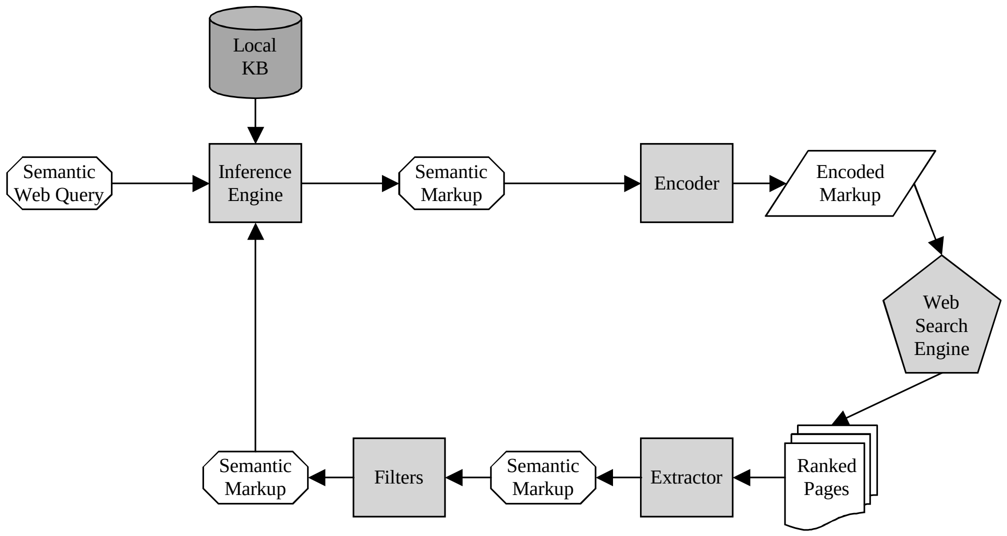

Figure 1. Integration of inference and retrieval over semantic markup. Arrows represent data flow.

ding RDF-based markup in HTML is non-compliant with HTML standards up to and including HTML 4.0. This issue is currently under study by a W3C task force [21].

The second way to bind HTML to semantic markup is to create a pair of documents, one containing HTML, the other containing the corresponding semantic markup. The two files are bound by placing in each a pointer to the URI of the other, either by URI naming convention, or by concurrent retrieval (*i.e.,* as part of a single transaction). While this method makes it difficult to associate semantic markup with specific components of the HTML page, it is possible to implement using today's standards.

Whichever approach is taken to binding semantic markup to HTML, the current lack of a standard has made it difficult to exploit the relationship between the two.

In the remainder of this paper, we first propose a framework for the integration of inference and retrieval. Next we describe OWLIR, a retrieval system that implements portions of the framework. We then explore some of the more difficult issues that must be resolved for full implementation of the framework, and provide concluding remarks.

2. OUR FRAMEWORK

One of the stated objectives of the semantic web is to enhance the ability of both people and software agents to find documents, information and answers to queries on the web. While there has been some research on information retrieval techniques applied to documents with markup [3,7,11], the role of explicit ontologies in information retrieval tasks [18], and on question answering as a retrieval task [16], much of it can be seen as incremental extensions to familiar paradigms. Our goal is more ambitious and offers, we think, a new paradigm for information retrieval that mixes and interleaves search, retrieval and understanding.

To explore the tight integration of search and inference, we propose a framework designed to meet the following desiderata:

- The framework must support both retrieval-driven and inference-driven processing.
- Retrieval must be able to use words, semantic markup, or both as indexing terms.
- Web search must rely on today's broad coverage, text-based retrieval engines.
- Inference and retrieval should be tightly coupled; improvements in retrieval should lead to improvements in inference, while improvements in inference should lead to improvements in retrieval.

In the following subsections, we first describe the portions of the framework that use semantic markup, then show how text processing can be mixed in to increase system capabilities and improve performance.

2.1 Processing of Semantic Markup

First, imagine we are concerned only with retrieval and inference over semantic markup. We would like the ability to operate some sort of inference engine, to identify facts and rules needed by the inference engine to reach its desired conclusions, to search the Semantic Web for such facts and rules, and to incorporate the results of the search into the inference process. Figure 1 shows the basic architecture of such a system.

Input to the system is some sort of Semantic Web query. If the user's goal is retrieval, this might simply be semantic markup

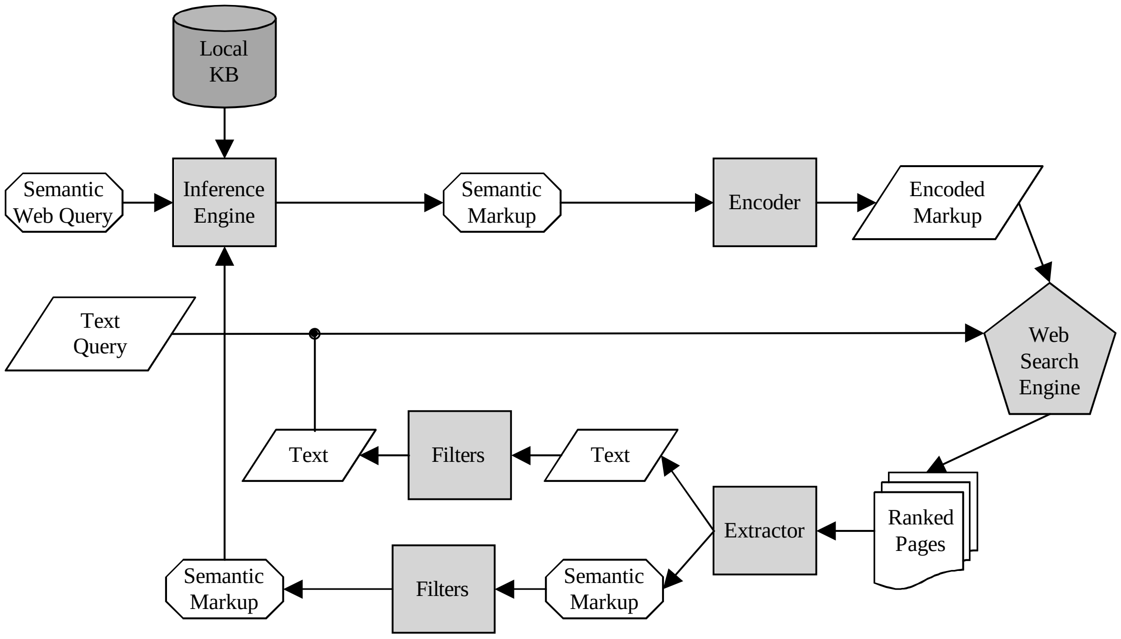

Figure 2. Text can also be extracted from the query results, filtered, and injected into the query.

encoding the concepts being sought (*e.g.*, using XML-QL [9] or XIRQL [13]). Alternatively, if the goal is inference, the query might be a statement the system is to prove. In either case, the query is submitted to the inference engine. For retrieval, the inference engine may choose to perform limited forward chaining on the input (as a text retrieval engine might perform thesaurus expansion). For proof, the inference engine will generate a partial proof tree (or more accurately, one in a sequence of partial proof trees), using its local knowledge base to the extent possible. The inference engine produces a description of the semantic markup to be sought on the Web.

Because we want to use a traditional Web search engine for the retrieval, we cannot simply use the output of the inference engine as a search query. Rather, we must first encode the semantic markup query as a text query that will be recognized by a search engine. We call this process *swangling*, for 'Semantic Web mangling.'[1] Technical details about swangling, and its application to Web pages prior to indexing, are discussed further below in Section 4. The result is a bag of words, recognizable as indexing terms by the target Web search engine(s), that characterize the desired markup.

The query is submitted to one or more Web search engines. The result will be a ranked list of Web pages, which either contain semantic markup themselves, or refer to companion pages that do. Some number of these pages must be scraped to retrieve their semantic markup. Control over how many pages to scrape, and over whether to scrape additional pages or to issue a new Web query, resides with the inference engine.

[1] Mangling is the technical term for a technique used in C++ and other object-oriented compilers in which the types of a method's arguments and return value are encoded in the internal function name.

Only some of the semantic markup retrieved through this process will be useful for the task at hand. Some will not come from an appropriate trusted authority. Some will be redundant. Some will be irrelevant. Thus, before it is asserted into the inference engine's knowledge store, the semantic markup gleaned from each page must be filtered. The result will be a collection of facts and rules, which are likely to further the inferences being pursued, or serve as valuable relevance feedback terms. These facts and rules are passed to the inference engine, which may then iterate the entire process.

2.2 Using Text

The process described in the previous subsection makes no use of text, except to the extent that the result of markup swangling is a set of text terms. However, there is no reason that we cannot include appropriate text in the Web query. Adding text will influence the ordering of search results, possibly biasing them toward pages that will be most useful for the task at hand. Figure 2 shows how text can be included in the framework. First, a text query can be sent directly to the search engine (augmented by swangled markup, if such is available). Second, the extractor can pull text as well as markup out of retrieved pages. As with semantic markup, extracted text may be filtered or transduced in various ways before being used. Potentially useful filters include translation, summarization, trust verification, *etc.*

Incorporation of extracted text into the query of a subsequent round of processing corresponds to blind relevance feedback. The framework therefore provides a way to include both text and semantic markup as relevance feedback terms, even when the original query is homogeneous.

3. AN EXAMPLE: OWLIR

OWLIR [22] is an implemented system for retrieval of documents that contain both free text and semantic markup in RDF, DAML+OIL or OWL. OWLIR was designed to work with almost any local information retrieval system and has been demonstrated working with two–HAIRCUT [20] and WONDIR. In this section we briefly describe the OWLIR system; readers are referred to Shah [22] for additional details.

While we have used OWLIR to explore the general issues of hybrid information retrieval, the implemented system was built to solve a particular task – filtering University student event announcements. Twice a week, UMBC students receive an email message listing 40-50 events that may be of interest, *e.g.,* public lectures, club meetings, sporting matches, movie screenings, outing, *etc.* Our goal was to automatically process these messages and produce sets of event descriptions containing both text and markup. These descriptions are then further processed, enriched with the results of local knowledge and inferencing and prepared for indexing by an information retrieval system. A simple form-based query system allows a student to enter a query that includes

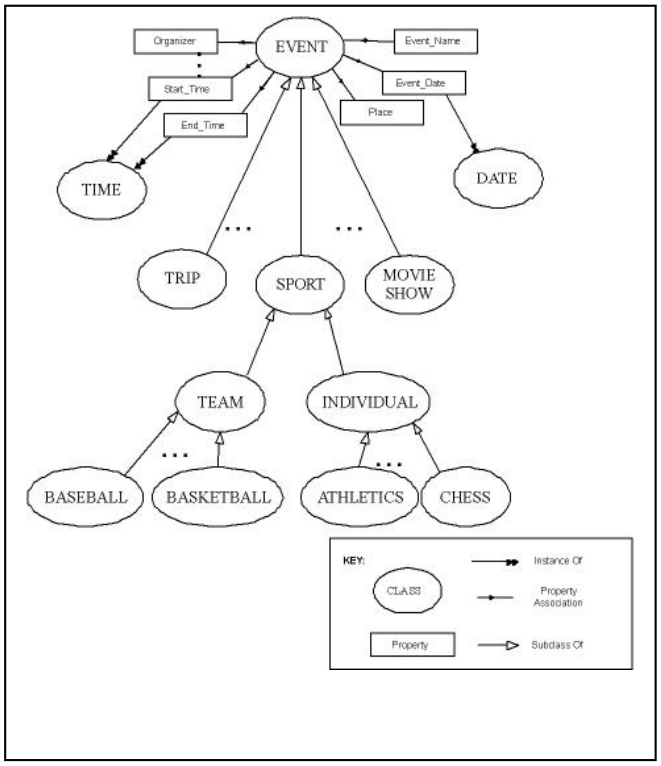

Figure 3. OWLIR annotations use terms from a DAML+OIL ontology of classes and properties that are useful in describing campus events.

both structured information (*e.g.,* event dates, types, *etc.*) and free text. The form generates a query document in the form of text annotated with DAML+OIL markup. Queries and event descriptions are processed by reducing the markup to triples, enriching the structured knowledge using a local knowledge base and inferencing, and swangling the triples to produce acceptable indexing terms. The result is a text-like query that can be used to retrieve a ranked list of events that match the query.

OWLIR defines ontologies, encoded in DAML+OIL, allowing users to specify their interests in different events. These ontologies are also used to annotate the event announcements. Figure 3 shows a portion of the OWLIR Event Ontology, which is an extension to the ontologies used in ITTalks [8]. Events may be academic or non-academic, free or paid, open or by invitation. An event announcement made within the campus is identified as an instance of one of the natural kind of events or subcategories. Instances of subcategories are inferred to be a subtype of one of the natural kind of events.

Text Extraction. Event announcements are currently in free text. We need these documents to contain semantic markup. We take advantage of the AeroText™ system to extract key phrases and elements from free text documents. Document structure analysis supports exploitation of tables, lists, and other elements to provide more effective analysis.

We use a domain user customization tool to fine-tune extraction performance. The extracted phrases and elements play a vital role in identifying event types and adding semantic markup. AeroText has a Java API that provides access to an internal form of the extraction results. We have built DAML generation components that access this internal form, and then translate the extraction results into a corresponding RDF triple model that uses DAML+OIL syntax. This is accomplished by binding the Event ontology directly to the linguistic knowledge base used during extraction.

Inference System. OWLIR uses the metadata information added during text extraction to infer additional semantic relations. These relations are used to decide the scope of the search and to provide more relevant responses. OWLIR bases its reasoning functionality on the use of DAMLJessKB [15]. DAMLJessKB facilitates reading and interpreting DAML+OIL files, and allowing the user to reason over that information. The software uses the SiRPAC RDF API to read each DAML+OIL file as a collection of RDF triples and Jess (Java Expert System Shell) [12] as a forward chaining production system to apply rules to those triples.

DAMLJessKB provides basic facts and rules that facilitate drawing inferences on relationships such as Subclasses and Subproperties. We enhance the existing DAMLJessKB inference capabilities by applying domain specific rules to relevant facts. For example, DAMLJessKB does not import facts from the ontology that is used to create instances, thereby limiting its capacity to draw inferences. We have addressed this issue by importing the base Event ontology and providing relevant rules for reasoning over instances and concepts of the ontology. This combination of DAMLJessKB and domain specific rules has provided us with an effective inference engine.

As an example of the swangling process used in OWLIR, consider the markup, expressed here in RDF N3 notation, describing a movie with the title "Spiderman":

_j:00255 a owlir:movie; dc:title "Spiderman".

OWLIR has domain-specific rules that are used to add information useful in describing an event. One rule is triggered by a description of a movie event where we know the movie title. This rule requests that the Internet Movie Database (IMDB) agent seek additional attributes of this move, such as its genre. The results are added as triples, such as the following one (also in N3).

_:j00255 owlir:moviegenre "action".

This triple is then expanded with wildcards to generate seven terms, which are added to the document prior to indexing:

1. j00255.owlir.umbc.edu/event/moviegenre.action
2. *.owlir.umbc.edu/event/moviegenre.action
3. j00255.*.action
4. j00255.owlir.umbc.edu/event/moviegenre.*
5. j00255.*.*
6. *.owlir.umbc.edu/event/moviegenre.*
7. **.action

We conducted experiments with OWLIR to see if semantic markup within documents can be exploited to improve retrieval performance. We measured precision and recall for retrieval over three different types of document: text only; text with semantic markup; and text with semantic markup that has been augmented by inference. We used two types of inference to augment document markup: reasoning over ontology instances (*e.g.,* deriving the date and location of a basketball game); and reasoning over the ontology hierarchy (*e.g.,* a basketball game is a type of sporting event). For example, extracting the name of a movie from its description allows details about the movie to be retrieved from the Internet Movie Database site. A query looking for movies of the type Romantic Genre can thus be satisfied even when the initial event description was not adequate for the purpose.

We generated twelve hybrid (text plus markup) queries, and ran them over a collection of 1540 DAML+OIL-enhanced event announcements.

Table 1. Mean average precision over twelve hybrid queries

Unstructured data (e.g., free text)	Structured data with inferred data	Structured data plus free text
25.9%	66.2%	85.5%

Indexed documents contain RDF Triples and RDF Triple Wildcards. This gives users the flexibility to represent queries with RDF Triple wildcards. DAML+OIL captures semantic relationships between terms and hence offers a better match for queries with correlated terms.

These experiments were run using the WONDIR information retrieval engine. Preliminary results are shown in Table 1 and in Shah *et al.* [22]. Retrieval times for free text documents and documents incorporating text and markup are comparable. Including semantic markup in the representation of an indexed document increases information retrieval effectiveness. Additional performance benefits accrue when inference is performed over a document's semantic markup prior to indexing. While the low number of queries at our disposal limits any conclusions we might draw about the statistical significance of these results, we are nonetheless strongly encouraged by them. They suggest that developing retrieval techniques that draw on semantic associations between terms will enable intelligent information services, personalized Web sites, and semantically empowered search engines.

4. DISCUSSION

A body of documents that contain both text and semantic annotations requires, or at least offers an opportunity to explore, new models for information retrieval that interleave document retrieval and inference. Such interleaving is done today, but is split between people and computers.

Here is a description of what occurs in a typical information retrieval session, focusing on the roles played by a person with a query (*e.g.,* "What is the capital of India") and the computer system(s) used to answer the query.

- A person mentally forms a semantic query;
- the person encodes the query as a combination of words and phrases that are thought to characterize documents that contain information needed to answer the query;
- the computer system retrieves a ranked set of documents matching the text query;
- the person reviews some of the highly ranked documents, reading and extracting some of their meaning;
- if the semantic query can now be answered, the process terminates with success; otherwise,
- Some of the newly extracted facts and knowledge are used to reformulate the text query, and the process repeats.

Our goal can be seen as attempting to completely automate this process. Achieving this goal will have two major benefits. First, for systems that start with a human-posed query, the person no longer needs to read and extract information from retrieved documents in order to answer the semantic query or reformulate the text query. Second, it allows software agents to extract knowledge from the Web to answer their semantic queries without the aid of a person.

Progress toward this goal will rely on solutions to a number of issues and open problems. The general categories include tokenization (mapping OWL onto word-like indexable tokens), reasoning (when and how much), trust and consistency (what sources to use), and dealing with search engines.

4.1 Tokenization

Most search engines are designed to use words as tokens. We have named the process of converting semantic web RDF triples to word-like tokens *swangling*. There are two immediate issues that present themselves – which triples to select for swangling and what techniques to use to swangle a selected triple.

What to swangle. Some search engines, such as Google, limit query size. Care must be taken to choose a set of triples that will be effective in finding relevant documents. Some triples carry more information that others. For example, every instance is a type of owl:thing, so adding triples asserting owl:thingness will not be very helpful, especially if the query size is limited. OWL and RDF descriptions typically contain anonymous nodes (also know as "blank nodes") that represent existentially asserted entities. Triples that refer to blank nodes should probably be processed in a special way, since including the "gensym" tag that represents the blank node carries no information. It might be possible to develop a statistical model for OWL annotations on documents similar to statistical language models. Such a model could help to select triples to include in a query.

How to swangle. In the OWLIR system we explored one approach to swangling triples. More experimentation is clearly needed to find the most effective and efficient techniques for reducing a set of triples to a set of tokens that a given information retrieval system will accept. The simplest approach would be to decompose each triple into its three components and to swangle these separately. This loses much of the information, of course. OWLIR followed an approach which preserved more information. Each triple was transformed into seven patterns, formed by replacing zero, one or two of its components with a special "don't care" token. Each of the seven resulting tokens was then reduced to a single word-like token for indexing.

4.2 Reasoning

When to reason. We have a choice about when to reason over Semantic Web markup. We can reason over the markup in a document about to be indexed, resulting in a larger set of triples. We can also reason over a query that contains RDF triples prior to processing it and submitting it to the retrieval system. Finally, we can reason over the markup found in the documents retrieved. In OWLIR, we chose to reason both over documents as they were being indexed and over queries about to be submitted. It is not obvious to us how much redundancy this entails nor is it clear if there is a best approach to when to do the reasoning.

How much to reason. A similar problem arises when one considers how much reasoning to do or whether to rely largely on forward chaining (as in OWLIR) or a mixture of forward and backward reasoning.

4.3 What knowledge to use

What to trust. The information found on the Semantic Web will vary greatly in its reliability and veracity, just as information on the current Web. It will not do just to inject into our reasoning the facts and knowledge from a newly found and relevant document. Moreover, we may need to take care not to create an inconsistent knowledge base. This problem is being studied in the context of models of trust on the Web [10][14][22].

Modeling document provenance. Much of the information found in a document comes from somewhere else – typically another document. Data provenance [6] is a term used for modeling and reasoning about the ultimate source of a given fact in a database or document. For systems that extract and reason about facts and knowledge found on the Semantic Web, it will be important to (i) inform our trust model and make better decision about the trustworthiness of each fact; and (ii) remove duplicate facts from our semantic model.

4.4 Dealing with search engines

Control. The basic cycle we've described involves (re)forming a query, retrieving documents, processing some of them, and repeating. This leaves us with a decision about whether to look deeper into the ranked result set for more information to use in reforming our query, or to reform the query and generate a new result set. The choice is similar to that faced by an agent in a multiagent system that has to frequently consider whether to continue reasoning with the information it has or to ask other agents for more information or to help with the reasoning [19]. We need some metric that estimates the expected utility of processing the next document in the ranked result set.

Spiders. Web search engines typically do not process markup. So, we need a way to give a search engine spider a preprocessed (swangled) version of a Web page when it tries to spider it for indexing. This can be easily accomplished if we have control of the HTTP server that serves a page – it checks to see if the requesting agent is a spider. If so, it returns the swangled version of the page, otherwise it returns the original source page. The preprocessing can be done in advance or on demand with caching.

Offsite annotation. The technique described above depends on having control over all of the servers associated with a Semantic Web page. If this is not the case, some work arounds are needed. One option is to mirror the pages on a server that does automatic swangling. The pages should have a special annotation (*e.g.,* in RDF) that asserts the relationship between the source and mirrored pages.

Search engine limitations. Web based search engines have limitations that must be taken into account, including how they tokenize text and constraints on queries. We would like swangled terms to be accepted as indexable terms. The two retrieval systems we used in OWLIR were very flexible in what they accepted as a token; tokens could be of arbitrary length and could include almost any non-whitespace characters. Many commercial systems are much more constrained. With Google, for example, we were advised to keep the token length less than 50 and to include only lower and uppercase alphabetic characters. Many commercial systems also limit the size of a query to a maximum number of terms. Google, for example, currently has a limit of ten terms in a query. These limitations, as well as others, affect how we have to interface to a given retrieval engine.

5. CONCLUSION

The Semantic Web will contain documents enriched by annotations that provide metadata about the documents as well as machine interpretable statements capturing some of the meaning of the documents' content. Information retrieval over collections of these documents offers new challenges and new opportunities. We have presented a framework for integrating search and inference in this setting that supports both retrieval-driven and inference-driven processing, uses both text and markup as indexing terms, exploits today's text-based Web search engines, and tightly binds retrieval to inference. While many challenges must be resolved to bring this vision to fruition, the benefits of pursuing it are clear.

6. REFERENCES

[1] Abiteboul, S., Quass, D., McHugh, J. Widom, J. and Wiener, J. 'The Lorel query language for semistructured data.' *International Journal on Digital Libraries* 1, pages 68-88, April 1997.

[2] Arocena, G. and Mendelzon, A. 'WebOQL: Restructuring documents, databases and webs.' In *International Conference on Data Engineering,* pages 24--33. IEEE Computer Society, 1998.

[3] Bar-Yossef, Z., Kanza, Y., Kogan, Y., Nutt, W. and Sagiv, Y.. 'Quest: Querying semantically tagged documents on the World Wide Web.' In *Proc. of the 4th Workshop on Next Generation Information Technologies and Systems*, volume NGITS'99, Zikhron-Yaakov (Israel), July 1999.

[4] Berners-Lee, T. and Fischetti, M. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor.* Harper, San Francisco. 1999.

[5] Berners-Lee, T., Hendler, J. and Lassila, O. 'The Semantic Web.' *Scientific American*, May 2001.

[6] Buneman, P., Khanna, S. and Tan, W-C. 'Why and Where: A Characterization of Data Provenance.' *International Conference on Database Theory (ICDT)* 2001.

[7] Chinenyanga, T. and Kushmerick, N. 'Elixir: An expressive and efficient language for XML information retrieval.' In *SIGIR Workshop on XML and Information Retrieval,* 2001.

[8] Cost, R. S., Finin, T., Joshi, A., Peng, Y., Nicholas, C., Soboroff, I., Chen, H., Kagal, L., Perich, F., Zou, Y., and Tolia, S. 'ITTALKS: A Case Study in the Semantic Web and DAML+OIL.' *IEEE Intelligent Systems* 17(1):40-47, 2002.

[9] Deutsch, A.,Fernandez, M., Florescu, D., Levy, A. and Suciu, D. 'XML-QL: A query language for XML.' In *Proceedings of the Eighth International World Wide Web Conference*, 1999.

[10] Ding, L., Zhou, L. and Finin, T. 'Trust Based Knowledge Outsourcing for Semantic Web Agents,' *2003 IEEE/WIC International Conference on Web Intelligence (WI 2003)*, October 2003, Halifax, Canada.

[11] Egnor, D. and Lord, R. 'Structured information retrieval using XML.' In *Proceedings of the ACM SIGIR 2000 Workshop on XML and Information Retrieval,* Athens, Greece, July 2000.

[12] Friedman-Hill, E. *Jess, the Java expert system shell.* Sandia National Laboratories. 2000.

[13] Fuhr, N. and Grojohann, K. 'XIRQL: An extension of XQL for information retrieval.' In *Proceedings of the ACM SIGIR 2000 Workshop on XML and Information Retrieval*, Athens, Greece, July 2000.

[14] Golbeck, J., Parsia, B., and Hendler, J. 'Trust networks on the Semantic Web.' To appear in *the Proceedings of Cooperative Intelligent Agents 2003*, August 27-29, Helsinki, Finland.

[15] Kopena, J. and Regli, W., 'DAMLJessKB: A tool for reasoning with the Semantic Web.' *IEEE Intelligent Systems* 18(3), May/June, 2003.

[16] Kwok, C., Etzioni, O. and Weld, D. 'Scaling question answering to the Web. ' In *Proceedings of WWW10,* Hong Kong, 2001.

[17] Martin, P. and Eklund, P. 'Embedding knowledge in Web documents.' In *Proceedings of World Wide Web Conference (WWW8),* Toronto, Canada, 1999.

[18] Mayfield, J. 'Ontologies and text retrieval.' *Knowledge Engineering Review* 17(1):71-75. 2002.

[19] Mayfield, J., Finin, T., Narayanaswamy, R., Shah, C., MacCartney, W. and Goolsbey, K. 'The Cycic Friends Network: Getting Cyc agents to reason together.' *Proceedings of the CIKM Workshop on Intelligent Information Agents*. 1995.

[20] Mayfield, J., McNamee, P. and Piatko, C. 'The JHU/APL HAIRCUT system at TREC-8.' *The Eighth Text Retrieval Conference (TREC-8),* pages 445-452, November 1999.

[21] Reagle, J. (ed.), *RDF in XHTML*. W3C Task Force Document, May 2003.

[22] Shah, U., Finin, T., Joshi, A., Cost, R. S. and Mayfield, J. 'Information Retrieval on the Semantic Web.' *10th International Conference on Information and Knowledge Management*, November 2002.