Information Retrieval On The Semantic Web*

Urvi Shah
TripleHop Technologies, Inc
New York, NY 10010
urvi@triplehop.com

Tim Finin
Dept. Comp. Sci.
University of Maryland
Baltimore County
Baltimore, MD 21227
finin@cs.umbc.edu

Anupam Joshi
Dept. Comp. Sci.
University of Maryland
Baltimore County
Baltimore, MD 21227
joshi@cs.umbc.edu

R. Scott Cost
Dept. Comp. Sci.
University of Maryland
Baltimore County
Baltimore, MD 21227
cost@cs.umbc.edu

James Mayfield
Johns Hopkins University
Applied Physics Laboratory
Laurel, MD 20732
james.mayfield@jhuapl.edu

ABSTRACT

We describe an approach to retrieval of documents that contain of both free text and semantically enriched markup. In particular, we present the design and implementation prototype of a framework in which both documents and queries can be marked up with statements in the DAML+OIL semantic web language. These statements provide both structured and semi-structured information about the documents and their content. We claim that indexing text and semantic markup together will significantly improve retrieval performance. Our approach allows inferencing to be done over this information at several points: when a document is indexed, when a query is processed and when query results are evaluated.

Keywords

Semantic Web, Text Extraction, Query-Answering Systems, Hybrid Information Retrieval

1. INTRODUCTION

We envision the future web as pages containing both text and semantic markup. Current information retrieval techniques are unable to exploit the semantic knowledge within documents and hence cannot give precise answers to precise questions. We cannot automatically extract such content from general documents yet. Manually structuring web documents, for example, with XML lets us retrieve more precise information using string and structure matching tools, such as the web robots Harvest, WebSQL, and WebLog. However, for this approach the user needs to be well aware of the structure of the documents, their exact names and forms, and hence is not scalable. Knowledge representation languages like DAML+OIL that support logic inferences can help us achieve more flexible and precise knowledge representation and retrieval. Industry is currently developing many metadata languages (e.g RDF(S), OML) to let people index web information resources with knowledge representations (logical statements) and store them in web documents. DAML+OIL is an effort to develop a universal Semantic Web markup language that is sufficiently rich to provide machines not only with the capability to read data but also with the capability to interpret and infer over the data.

Web documents may contain free text along with mark up. There are many potential uses for annotation on the semantic web including workflow, image retrieval, database mediation and device interoperability. We have developed a system *OWLIR (Ontology Web Language and Information Retrieval* which focuses on addressing three scenarios that involve semantically marked up web pages and text documents:

Information retrieval (IR) - e.g., identify and rank relevant pages or documents for a query looking for detail descriptions concerning USA and Afghanistan leaders.

Simple question answering (Q&A) - e.g., who is the president of the USA?

Complex question answering - e.g., what is the current situation in Afghanistan?

We have several general techniques to improve retrieval efficiency and performance. All three scenarios involve some degree of reasoning and inference. We present preliminary results for a defined set of queries and documents for the first two scenarios. Complex Q&A will often involve significant reasoning and summarization capabilities and our system sets the grounds for the same.

*This research was supported in part by DARPA contract F30602-97-1-0215.

CIKM'02, November 4–9, 2002, McLean, Virginia, USA.

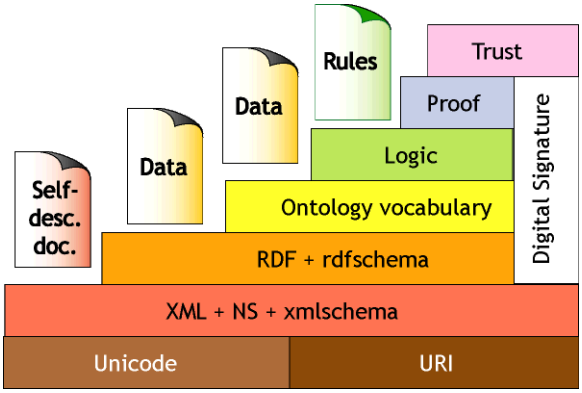

Figure 1: Tim Berners Lee's vision of the Semantic Web

2. BACKGROUND

2.1 DAML and the Semantic Web

The current web is primarily composed of pages with information in the form of natural language text and images intended for humans to view and understand. Machines are used primarily to render this information, laying it out on the screen or printed page. The idea behind the Semantic Web is to augment these pages with markup that captures some of the meaning of the content on pages and encodes it in a form that is suitable for *machine understanding* [18]. This requires a new kind of markup languages; one that supports defining shared data models or ontologies for a domain and allows page authors to make statements using this ontology. The markup languages currently being used to explore this idea include RDF/S [26, 22] and DAML+OIL [11, 12].

The XML standard [7] provides the necessary means to declare and use simple data structures, which are stored in XML documents and which are machine-readable. However, since XML is defined only at the syntactic level, machines cannot be relied upon to unambiguously determine the correct meaning of the XML tags used in a given XML document. The W3C Consortium has developed RDF/S with the goal of addressing the XML deficiencies by adding formal semantics on the top of XML. These standards are still very restricted as a knowledge representation language due to the lack of support for variables, general quantification, rules, etc.

The goal of DAML+OIL is to enable the transformation of the currently human-oriented web, which is largely used as a text and multimedia repository only, into a Semantic Web as envisioned by Berners-Lee [5, 6]. It follows the same path for representing data and information in a document as XML, and provides similar rules and definitions to RDF/S. In addition, DAML+OIL also provides rules for describing further constraints and relationships among resources, including cardinality, domain and range restrictions, as well as union, disjunction, inverse and transitivity rules. DAML+OIL will enable the development of intelligent agents and applications that can autonomously retrieve and manipulate information on the Internet and from the Semantic Web of tomorrow.

2.2 Information Retrieval in the World Wide Web

Though an active area of research for over thirty years, information retrieval (IR) has only become ubiquitous with the advent of the World Wide Web. The most familiar application of text retrieval is ad-hoc querying where a query is used to search a static set of documents. This is the task that commercial web search engines such as AltaVista and Google are best known for addressing. Search engines operate on huge databases and carry out a keyword search. In most cases precision is low, not all retrieved documents answer a user's query. For example, when a query "Who is the President of USA", was posed to Google, few links retrieved contained the name of the current President somewhere in the document, but documents describing "How to campaign for becoming a President" and the President of a newspaper "USA TODAY" were also retrieved.

Intelligent Search Engines evolved as a descendent to Meta Search Engine, which incorporate machine-learning techniques. Knowledge can be annotated on the page in such a way that automatic tools can collect and understand it, enabling intelligent information services, personalized web sites, and semantically empowered search engines.. Ontologies make possible that software agents can understand knowledge, which is marked up and further draw inferences pertaining to the domain of interest[28].

Agent Paradigm is a promising technology for information retrieval. Some applications are intelligent IR interfaces and clustering and categorization. An agent-based approach means that IR systems can be more scalable, flexible, extensible, and interoperable. Agents need a way to process and "understand" their information, both on the level of individual documents/objects as well as collection-wide entities. Statistical approaches, for deriving metadata from information, such as n-grams and latent semantic indexing are particularly interesting for analyzing text objects, because they are independent of the language of the text, are resistant to misspellings, and allow the application of many known mathematical techniques to natural language analysis.

2.3 Answering queries on the Web

Query-Answering Systems has been an area of research in different fields such as knowledge representation, databases, information retrieval, and natural language. The advantage of these precision based system coupled with the scope of search engines led to efforts in scaling these systems to the web. START [19] is one of the first QA systems with a web interface, having been available since 1993. Focused on questions about geography and the MIT InfoLab, START uses a precompiled knowledge base in the form of subject-relation-object tuples, and retrieves these tuples at run time to answer questions. AskJeeves [3] is a commercial service that provides a natural language question interface to the web, but it relies on hundreds of human editors to map between question templates and authoritative sites. MULDER [21] utilizes several natural-language parsers and heuristics in order to return high-quality answers. Using this framework the MULDER system can be modeled to suit the requirements of answering queries on the web.

2.4 DAML queries

The purpose of ontologies and annotations on web pages is to enable a level of query capability and performance that is

not available with current web search technology. RDQL is an implementation of an SQL-like query language for RDF. It treats RDF as data and provides query with triple patterns and constraints over a single RDF model. TRIPLE is an RDF-based logic programming language for the Semantic Web at the Stanford University Database Group. TRIPLE's [27] layered architecture allows simple object-oriented extensions like RDF Schema, directly implemented with the extended Horn logic features and DAML+OIL type modules can be realized via interaction with external reasoning components.

DQL, a DAML+OIL Query Language, is a simple language for querying DAML+OIL knowledge bases. The language is specified as DAML+OIL ontology so that both queries and the results obtained from asking a query are represented in DAML+OIL. In order to query a DAML+OIL KB, one expresses a query in DAML+OIL and the results are returned in DAML+OIL. An instance of class Query represents a question posed to a reasoner. A query pattern is in affect a conjunction of one or more triples. Each triple corresponds to an RDF Statement except that its predicate, subject, and/or object can be a variable.

3. RELATED WORK

The web is currently a distributed mass of simple hypertext documents. Large-scale web search engines effectively retrieve entire documents, but they are imprecise, because they do not exploit semantics of content. WebKB [23] is a tool that interprets semantic statements stored in web-accessible documents. WebKB advocates the use of Conceptual Graphs and simpler notational variants for ontology and control commands that enhance knowledge readability and let its users combine lexical, structural, and knowledge-based techniques to exploit or generate web documents. In an operational context, these knowledge-based features need to be combined with more traditional information retrieval ideas that give both coarse-grained search capabilities and the fine-grained, precision-based knowledge retrieval. WebOQL [2] provides a framework that supports a large class of data restructuring operations. It serves as a two-way bridge between databases and the Web and supports applications like Web-data warehousing.

Quest [4] was designed and implemented for querying and manipulating documents written in the OHTML markup language. OHTML supports fine granularity semantic tagging of HTML pages. Quest uses the W3Lorel query language, based on the Lorel [1] language to query the OEM objects (semantic view), as well as the hypertext view (HTML tags) of the document. Following this semi-structured approach, Quest allows for arbitrary tagging of HTML pages, offering flexibility to the user on the choice of semantic tags.

The SIGIR workshop on XML and information retrieval targeted various issues, which define the most relevant topics in the relation between these two technologies. ELIXIR [8], an Expressive and Efficient Language for XML Information Retrieval, extends the query language XML-QL [13] with a textual similarity operator. Based on the document-centric view of XML, XIRQL [17] is an extension of XQL that supports IR-related features, which are weighting and ranking, relevance-oriented search, data types with vague predicates, and semantic relativism. XIRQL integrates these features by using ideas from logic-based probabilistic IR models, in combination with concepts from the database area. XYZFind [14] is a system for structured information retrieval using XML. It incorporates techniques for exploiting semantically structured XML to increase precision and recall and an extension to the classic inverted index to support structured Information Retrieval.

4. DESIGN AND IMPLEMENTATION OF OWLIR

There is a fundamental conflict between a person's view of the Semantic Web and a software agent's view of it that must be resolved for the Semantic Web to adequately support retrieval. People will tend to view them as text documents that happen to contain some additional information that is not directly accessible or useful to them. Software agents on the other hand will view them as propositional stores over which to perform inference. These disparate views are incompatible on the surface. If they are not reconciled, they may lead to the development of a Semantic Web that is divorced from the current human-accessible Web.

To draw these disparate views together, and thereby increase the value of markup for people and the value of text for software agents, we argue that search and inference should be tightly bound. People will want to use the semantic Web to search not just for documents, but also for information about specific semantic relationships. Doing so will naturally require that inferences be drawn along the way. Software agents want to draw inferences on a topic of interest. Yet, because it is no longer practical to assume a monolithic knowledge base, drawing appropriate inferences necessitates rules and facts that will support the desired inferences.

There is a wide spectrum of techniques, which can be applied to address querying, and retrieval of semantically marked documents. OWLIR is intended to provide a framework, which is able to extract and exploit the semantic information from these documents, perform sophisticated reasoning and filter results for better precision.

OWLIR is an information retrieval system for Semantic Web documents, currently in the context of event announcements in the University domain. OWLIR can be described in terms of two primary components: a set of ontologies and a hybrid information retrieval mechanism. 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. The information retrieval engine is based on the use of WONDIR[9]. The framework employs text extraction, annotation, and inferencing mechanism, by utilizing the knowledge expressed in the ontologies.

There is a wide spectrum of techniques, which can be applied to address querying, and retrieval of semantically marked documents. OWLIR can help bypass some of the limitations of information access:

- Use semantic information for guiding the query answering process.
- Enable answers with a well-defined syntax and semantics that can directly be understood and further processed by automatic agents or other software tools.
- Provide information that is not directly represented as facts in the WWW, but which can be derived from other facts and some background knowledge.

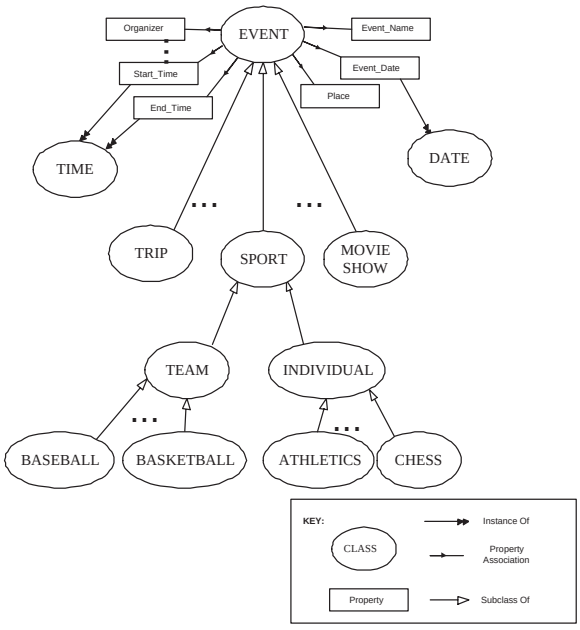

Figure 2: Snapshot of the Event Ontology

OWLIR is intended to provide a framework, which is able to extract and exploit the semantic information from DAML+OIL marked up documents, perform sophisticated reasoning and filter results for better precision. We now present the design and implementation aspects of every component of the framework.

4.1 EVENT Ontology

The term *Ontology* has been used in several disciplines, from philosophy, to knowledge engineering, where ontology is comprised of concepts, concept properties, relationships between concepts and constraints. Ontologies are defined independently from the actual data and reflect a common understanding of the semantics of the domain of discourse. Ontology is an explicit specification of a representational vocabulary for a domain; definitions of classes, relations, functions, constraints and other objects. Pragmatically, a common ontology defines the vocabulary with which queries and assertions are exchanged among software entities. Ontologies are not limited to conservative definitions, which in the traditional logic sense only introduce terminology and do not add any knowledge about the world. To specify a conceptualization we need to state axioms that put constraints on the possible interpretations for the defined terms.

Before we can annotate our Web pages with semantic information, we need to first establish one or more ontologies or use already-established ones that define how we can classify our documents. These ontologies describe categories, which our pages can fall into, and relationship rules between categories or other data, which we can use later to describe relationships between our pages and other data like numbers or dates.

DAML allows us to specify ontologies and markup pages to facilitate automatic knowledge extraction. The main goal of our ontology development was to develop an ontology, which will help users interested in different events in the university, retrieve relevant information. The *Event Ontology*, an extension to ITTalks [10] is built, following the concept of "Natural Kinds OF" [25] from the field of philosophy. Natural kind terms are clearly significantly different from many other general terms of natural languages. The reason for their peculiarity can be sought either in the kind of relation that holds between them and their semantic contents or in their contents themselves. We first identify the natural kinds in the phenomena under study, "EVENTS", and then figure out what their most important characteristics are. The Event ontology we built is within a University domain, which in turn is used to formulate semantic queries and to deliver exactly the information we are interested in. Event categories follow the natural kind of events that are prominent in a university e.g. Movie Showing, Seminars, Sport events etc. Every event has common properties like Date and Time of event, Organizer and Place of the event etc. Events may be academic or non academic, free or paid, open or by invitation, but these describe the events in general and are not identifying characteristics of any particular type of event. Figure 2 sketches a snapshot of the Event Ontology its categories and properties. 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. For the movie announcements we have chosen to create an ontology for the popular Internet Movie Database(IMDB). The *IMDB Ontology* is structured based on the IMDB. Almost all data about a movie that can be displayed to the end user is incorporated into the ontology.

A peculiar aspect of OWLIR is that the ontology not only affects how germane knowledge is retrieved, but also influences system interaction, at the encoding and end-user levels. The ontology becomes a means of communication between the user and the system and helps overcome the bottlenecks in information access, which is primarily based on keyword searches. It supports information retrieval based on the actual content of a page and helps navigate the information space based on semantic concepts. Ontologies enable advanced query answering and information extraction service, integrating heterogeneous and distributed information sources enriched by inferred background knowledge.

4.2 Information Extraction

Information extraction involves extracting specific types of task dependent information according to predefined guidelines. Event announcements are currently in free text. We need these documents to contain semantic markup. NLP-based extraction techniques as opposed to simple keyword, proximity or topic/entity searches are needed for reasonably accurate extraction for this task. We take advantage of the AeroText™ system for text extraction of key phrases and elements from free text documents. Document structure analysis supports exploitation of tables, lists, and other elements and complex event extraction to provide more effective analysis.

We use the Aerotext™ domain user customization tool to fine-tune extraction performance. The extracted phrases and elements play a vital role in identifying type of events and adding semantic markup. The AeroText™ system has a Java API that is used to access an internal form of the extraction results. We have improved AeroText™ by building DAML generation components that access this internal

form, and then translate the extraction results into a corresponding RDF triple model that utilizes the DAML+OIL syntax. This is accomplished by referencing the Event ontology that directly correlates to the linguistic knowledge base used in the extraction process.

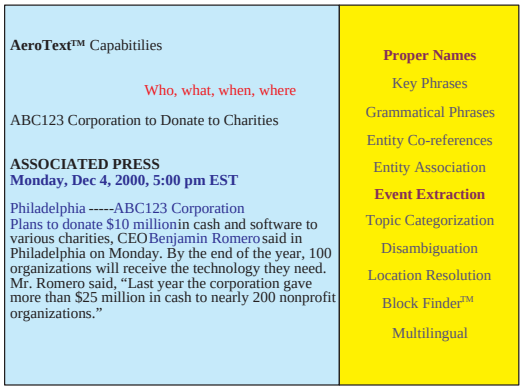

Figure 3: AeroText™ Capabilities

4.3 Inference System

OWLIR uses the metadata information added during the text extraction process to infer additional semantic relations. These relations are used to decide the scope of the search and to provide more germane responses. The inference engine exploits two information sources for deriving an answer: Event Ontology and the facts in the knowledge base. OWLIR bases its reasoning functionality on the use of DAMLJessKB [20]. DAMLJessKB facilitates reading DAML+OIL pages, interpreting the information as per the DAML+OIL language, and allowing the user to reason over that information. The software employs the SiRPAC RDF API to read in the DAML+OIL file as a collection of RDF triples and Jess (Java Expert System Shell) [16] as a forward chaining production system to carry out the rules of the DAML+OIL language. Jess is a rule engine and scripting environment written in the Java language that can be used to write applications that have the capacity to "reason" using knowledge supplied in the form of declarative rules. Jess uses the Rete [15] algorithm to process rules, a very efficient mechanism for solving the difficult many-to-many matching problem.

DAMLJessKB provides basic facts and rules that facilitate drawing inferences on relationships such as Subclasses and Subproperties. We enhance the existing inferential capabilities of DAMLJessKB and supplement it by filtering out facts that are of relevance to our system and applying domain specific rules. 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.

4.4 Information Retrieval System

The Hopkins Automated Information Retriever for Combing Unstructured Text (HAIRCUT) [24] is a information retrieval system we have developed at the Johns Hopkins University Applied Physics Laboratory (JHU/APL). A language modeling approach to reasoning document similarity is used in lieu of traditional Boolean or vector-space models, a variety of tokenization schemes is supported, including overlapping character n-grams, and a novel term similarity measure is used to support various linguistic operations. The system is implemented in Java for ease of development and portability. We have further enhanced HAIRCUT for indexing DAML+OIL and RDF Triples, with or without wildcards. HAIRCUT allows the user to specify required, allowed and disallowed query terms. This gives the user flexibility in querying, at the same time increases precision. The combination of several complementary technologies in a single system makes HAIRCUT distinctive among retrieval systems.

We have used WONDIR, built at UMBC, as the information retrieval engine for OWLIR. WONDIR (Word Or N-gram based Dynamic Information Retrieval Engine) [9] is a information retrieval engine written completely in Java. It provides basic indexing, retrieval and storage facilities for documents. Its main functions include ability to index terms as N-grams or as plain language words as the need be. It implements the standard cosine similarity metric to process free text queries. WONDIR's features include ability to handle large dynamic corpora and relative ease of usage.

4.5 Hybrid Information Retrieval

The addition of semantic markup to Web documents makes it possible to perform inference over document content. However, markup is also useful in another way. Traditional text retrieval characterizes documents by the indexing terms they contain. These indexing terms are typically words, but they need not be. One common variant is stemmed words; stems are words that have had suffixes removed to allow similar words (e.g.,juggler and juggling) to be treated as a single indexing term. Less commonly used, but no less powerful, are characters n-grams, overlapping sequences of n contiguous characters. The efficacy of these different types of indexing terms demonstrates that traditional approaches to text retrieval can be effective over a variety of term types. This suggests that semantic markup, if present, might serve as indexing terms for a traditional IR engine. That is, in addition to indexing documents according to the text of their words, stems or n-grams, we might also index them according to the text of their semantic markup. We could, for example, treat each distinct DAML+OIL tag as an indexing term. Or, we might reduce document markup to RDF triples, and treat each distinct triple as an indexing term; this is our current approach.

By including semantic markup as indexing terms, we are exploiting the statistical associations between semantic markup and text. For example, given a way to find strongly associated indexing terms (e.g., through mutual information), we could suggest markup for a word or phrase by finding semantic markup that is strongly associated with it in an indexed collection. Alternatively, we could identify text that characterizes a given markup tag or triple that might serve as a basis for automated or semi-automated ontology mapping.

5. OWLIR PROCESS FLOW

A retrieval engine sits on top of the document collection and handles retrieval requests. Information about events

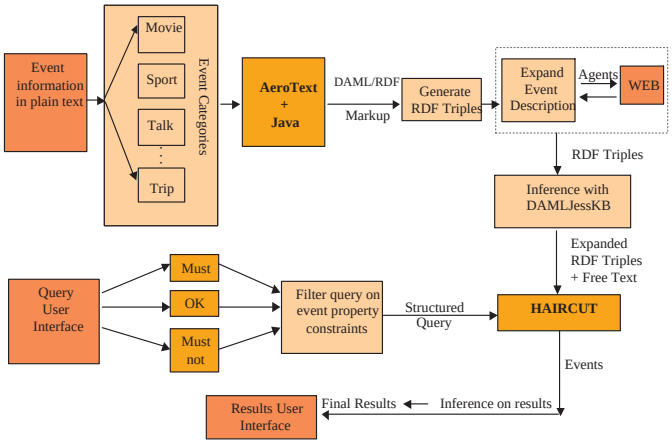

Figure 4: OWLIR Process Flow

in the university is collected for retrieval and analyzed for "Natural Kinds of Events". To make possible analysis and understanding of meaning of data by software, an extraction and population phase uses AeroText™ and the Event ontology to describe the documents in DAML+OIL. Documents containing RDF triple patterns generated from the DAML+OIL markup and triples that are inferred through DAMLJessKB inference system form the knowledge base of the information retrieval system. The IR engine preprocesses and indexes these documents. DAML+OIL being a machine understandable language, software agents can assist in document expansion. As an instance knowing the name of a Movie from the description of movie showing event announcements, an agent can gather further information about the movie from the IMDB site and detail the movie-showing event sketch. A set of DAML+OIL documents can be regarded as a database and can be directly processed by an application or queried via query languages for XML, DAML.

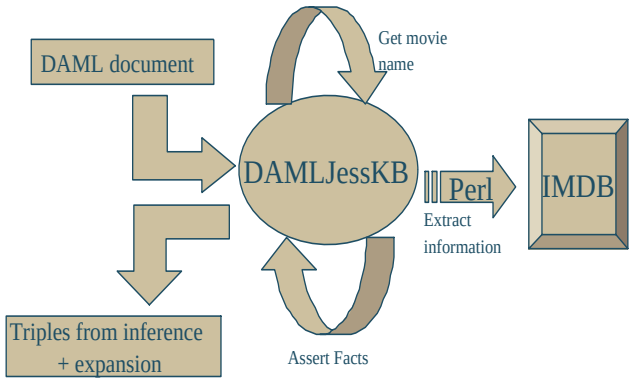

Figure 5: Information from IMDB Site

A query formulation mechanism translates natural language questions into queries for the IR engine in order to retrieve apposite documents from the collection, i.e., documents that can potentially answer the question. Keeping different kinds of users who have varied preferences, we built a set of DAML queries. A query pattern is in affect a conjunction of one or more triples. Each triple corresponds to an RDF Statement except that its predicate, subject, and/or object can be a variable. Queries can use rich data types, including numeric attributes, geographic location, temporal values and other quantities whose semantics are difficult to capture with keyword search. Query provides one way in which the programmer can write a more declarative statement of what is wanted, and have the information retrieval system retrieve it. Taking advantage of the HAIRCUT feature which allows the user to specify which terms in the query MUST, MUSTNOT and MAYBE considered, each query is expressed as a document consisting of triples and free text. Syntactically an XML markup, the query pattern identifies the necessary and sufficient conditions for the search. Refer to Figure 6 for Document Structure.

```
<DOC>
<DOCNO>'http://gentoo.cs.umbc.edu/howlir/announcements/charity#charity_001
</DOCNO>
<TEXT>'UMBC Blood Drive!!
Office of Student Life launches its annual Blood Drive for the Red Cross
on Mon, Nov 20 in the UC Ballroom from 10am - 4pm. </TEXT>
<TRIPLE>triple(charity_001)(
 'http://gentoo.cs.umbc.edu/howlir/announcements/charity#charity_001_place',
 'http://gentoo.cs.umbc.edu/ontologies/event_ont#Building',
 'University Center').
triple(charity_001)(
 'http://gentoo.cs.umbc.edu/howlir/announcements/charity#charity_001',
 'http://gentoo.cs.umbc.edu/ontologies/event_ont#Organizer',
 'Office of Student Life').
triple(charity_001)(
 'http://gentoo.cs.umbc.edu/howlir/announcements/charity#charity_001_date',
 'http://gentoo.cs.umbc.edu/ontologies/event_ont#Day_of_week',
 'Monday').</TRIPLE>
</DOC>
```

Figure 6: Document Structure

Logical systems provide provably good answers, but don't scale to large problems; an aspect that search engines can handle remarkably well. On the Semantic Web we can imagine a combination of a logical system coupled with the information retrieval engine. During search operation the retrieval system retrieves all the documents that reference the terms used in the query, and then a logical system can act on that closed finite world of information to determine a reliable solution if one exists.

6. EXPERIMENTAL ANALYSIS

The aim of our experiments is to verify that semantic markup within documents can be exploited to achieve better retrieval performance. We wanted to measure the extent to which Precision and Recall (P/R) is improved with the use of semantic markup. The baseline measurements are the P/R values obtained from the free text documents. 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 also ran queries over documents enhanced by means of gathering information from external sources. We built a "test" collection, which is a collection with available query relevance judgments for all queries. We evaluated our system using the TREC evaluation package available from the TREC Web site.

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 Event is a subclass of Sport Event). DAML+OIL being a machine understandable language, software agents can assist

Unstructured Data	*Structured Data+ Free Text*	*Structured+ Inferred Data+ Free Text*
25.86%	66.15 %	85.48 %

Table 1: 11-pt Average Precision

in document expansion, which facilitates answering queries beyond the scope of only free text. Document expansion, resulting from an external source in many ways facilitates answering queries beyond the scope of only free text. As an instance knowing the name of a Movie from the description of movie showings, an agent can gather further information about the movie from the IMDB site and detail the movie-showing event sketch. A query looking for movies of the type Romantic Genre can thus be satisfied although the initial event description was not adequate for the purpose. The system can now answer precise queries, comprising of triples and text, presented by the user. We generated 50 hybrid (text plus markup) queries and ran them over a collection of 2540 DAML+OIL-enhanced event announcements.

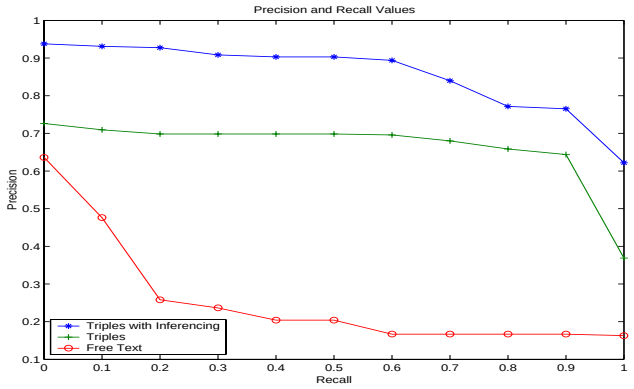

Figure 7: Precision and Recall Values

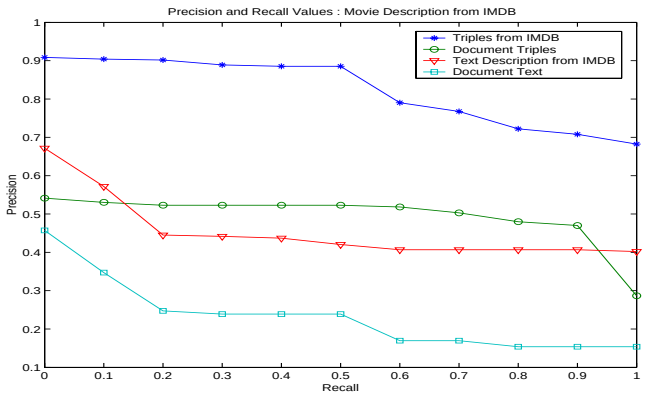

Figure 8: Precision and Recall Values - Enhanced Triples from IMDB

6.1 Discussion

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

Imposing a simple structure and semantics on both queries and data encoding can obtain substantial precision increases, in conjunction with adopting sense matching instead of word matching resulting in higher recall.

Our experiments were run using WONDIR information retrieval engine and preliminary results are shown in Table 1, Table 2 and Figure 7, Figure 8. At lower recalls the curves representing experiments with triples show higher precision values than the regular text documents. This means that a greater number of relevant documents was retrieved and ranked higher than those with the text documents. We attribute this to the inference capabilities and indexing of text and triples as a means of hybrid information retrieval. It is evident that semantic markup contained in documents and additional information from external sources, beget higher precision.

Retrieval times for free text documents and documents incorporating text and markup are comparable and time for indexing over DAML markup and inferencing can be amortized by preprocessing steps done offline. We believe that performance in terms of speed is not as important in this case as performance in terms of what is retrieved. Including semantic markup in the representation of an indexed document proves ostensibly valuable for information retrieval. Additional performance benefits accrue when inference is performed over a document's semantic markup prior to indexing. While the low number of documents and queries at our disposal limits any conclusion we might draw about the statistical significance of these results, we are nonetheless strongly encouraged by these results. This invariably sets a direction for developing retrieval techniques that draw on semantic associations between terms enabling intelligent information services, personalized Web sites, and semantically empowered search engines.

7. CONCLUSION AND FUTURE WORK

In this paper we have presented an approach to information retrieval over the Semantic Web utilizing a set of ontologies and inference engine. DAML+OIL is used as the knowledge representation language and as an interface for the inference engine, thus fostering flexibility and interoperability. The powerful support for rules formulation, constraints and question answering over schema information surpasses what is available in existing database technology. Inference service can be used to answer queries about explicit and implicit knowledge specified by the ontology thus providing a query answering facility that performs deductive retrieval from knowledge represented in DAML+OIL. Indeed, the retrieval of precise information is better supported by languages designed to represent semantic content and support logical inference, while the readability of such

	Movie Announcements	**Movie Description from IMDB**
Triples	49.28%	80.31%
Free Text	23.35%	45.60%

Table 2: 11-pt Average Precision - Enhanced Description from IMDB

a language eases its exploitation, presentation and direct insertion within a document (thus also avoiding information duplication). The OWLIR framework advocated the interdependency of search and inference for precise retrieval over semantic content. Thus OWLIR is an integrated comprehensive system to extract, reason and generate semantic markup employing the lexical, structural and knowledge-based approaches, which are complementary for information retrieval and exploitation. Our initial work on this framework is promising, and we have a prototype within the context of university events. The scalable knowledge repository we are building will permit the fusion and reuse of knowledge from various sources. In an operational context, these knowledge-based features need to be combined with more traditional information retrieval ideas that give both coarse-grained search capabilities and the fine-grained, precision-based knowledge retrieval we have described here.

Our ongoing work is to build a sophisticated inference engine, which can enhance a collection from implicit and explicit inferences made from semantic markup. The framework encourages the use of agents to route information and share metadata, facilitate document and query expansion, and evaluate query results. The OWLIR framework can be expanded to realize intelligent message routing, wherein event announcements can be routed to users based on their preferences, which are expressed as user profiles. User feedback can then be exploited for better precision.

8. REFERENCES

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

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

[3] Askjeeves. http://www.askjeeves.com.

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

[5] T. Berners-Lee and M. Fischetti. *Weaving the web: The original design and ultimate destiny of the World Wide Web by its inventor*. Harper, San Francisco.

[6] T. Berners-Lee, J. Hendler, and O. Lassila. *The Semantic Web*. Scientific American, May 2001.

[7] T. Bray, J. Paoli, and C. Sperberg-McQueen. Extensible markup language (xml). *W3C (Worldwide Web Consortium)*, 1998. http://www.w3.org/TR/1998/REC-xml-19980210.html.

[8] T. Chinenyanga and N. Kushmerick. Elixir: An expressive and efficient language for xml information retrieval. *In SIGIR Workshop on XML and Information Retrieval*, 2001.

[9] R. Cost. Wondir, word or n-gram based dynamic information retrieval engine. www.cs.umbc.edu/ cost/sire.

[10] R. Cost, T. Finin, A. Joshi, Y. Peng, C. Nicholas, H. Chen, L. Kagal, F. Perich, Y. Zou, and S. Tolia. ITTALKS: A Case Study in the Semantic Web and DAML. In *International Semantic Web Working Symposium (SWWS)*, July 2001.

[11] Darpa agent markup language, 2001. http://www.daml.org.

[12] DAML+OIL Design Rationale, 2001. www.cs.man.ac.uk/horrocks/Slides/index.html.

[13] A. Deutsch, M. Fernandez, D. Florescu, A. Levy, and D. Suciu. Xml-ql: A query language for xml. In *In Proc. 8th Int. World Wide Web Conference*, 1999.

[14] D. Egnor and R. Lord. Structured information retrieval using xml. XYZFind Corporation, Washington, USA.

[15] C. Forgy. Rete: A fast algorithm for the many object pattern match problem. *Artificial Intelligence*, 19:17–37, 1982.

[16] E. Friedman-Hill. Jess, the java expert system shell, 2000. http:/herzberg.ca.sandia.gov/jess/.

[17] N. Fuhr and K. Grojohann. Xirql: An extension of xql for information retireval. *In SIGIR Workshop on XML and Information Retrieval*, 2000.

[18] J. Heflin, J. Hendler, and S. Luke. Shoe: A prototype language for the semantic webs. *Linkvping Electronic Articles in Computer and Information Science*, 6 2001. http://www.ep.liu.se/ea/cis/1997/013/.

[19] B. Katz. From sentence processing to information access on the world wide web. *In Natural Language Processing for the World Wide Web*, pages 77–94, 1997. Papers from the 1997 AAAI Spring Symposium.

[20] J. Kopena. http://plan.mcs.drexel.edu/projects/legorobots/design/software/damljesskb/.

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

[22] O. Lassila and S. R. R. (eds). Resource description framework (rdf) model and syntax specification. *W3C Recommendation*, February 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.

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

[24] J. Mayfield, P. McNamee, and C. Piatko. The jhu/apl haircut system at trec-8. *The Eighth Text Retrieval Conference (TREC-8)*, pages 445–452, November 1999.

[25] W. V. Quine. *Naming, Necessity and Natural Kinds*, chapter Natural Kinds. University Press, 1977.

[26] *Resource Description Framework (RDF) Model and Syntax Specification*, February 1999. www.w3.org/tr/rec-rdf-syntax.

[27] M. Sintek and S. Decker. Triple-an rdf query, inference, and transformation language. *DDLP*, October 2001. Japan.

[28] S. Staab, M. Erdmann, A. Maedche, and S. Decker. An extensible approach for modeling ontologies in rdf(s). Technical Report 401, AIFB, University of Karlsruhe, March 2000.