The Anatomy and Design of A Semantic Search Engine

Wang Wei, Payam M. Barnaghi, Andrzej Bargiela

School of Computer Science
University of Nottingham (Malaysia Campus)
Jalan Broga, 43500 Semenyih, Selangor Darul Ehsan, Malaysia

Abstract

In recent years considerable research efforts have been devoted to applying semantic web technologies into information search and retrieval process. There are a number of pilot semantic search projects and frameworks being implemented and evaluated in various application domains. A survey of these systems and frameworks is necessary to gain a an overall view on current status of the research. This article serves two main purposes: the first part provides a survey on some of the existing semantic search systems focusing on their commonalities, objectives, scopes, methodologies, and technologies involved. We then demonstrate the IRIS system, a prototype semantic search system which helps researchers to search and explore collections of large number of scientific publications. The architecture and components of the system are described through the paper. We discuss our experiences in developing the IRIS system and describe issues with regard to current and future research in the semantic search area.

Key words: Semantic Search, Semantic Web, Ontology, Information Retrieval, Semantic Search Engine

1. Introduction

The World Wide Web has grown to an unprecedented scale and is still growing at a significant pace. The resulting wealth of information hinders people from locating relevant information quickly on one hand, while prompts the development of information search and retrieval technologies on the other hand. Powerful search engines built upon conventional search technologies have facilitated finding useful information easily and quickly on the Web. Nevertheless, the shortcomings of these conventional techniques have been recognised and discussed in large amount of literature, for example, it is often the case that people need to query search engines several times and combine the results to extract final answers. The semantic web [4] is being developed based on the current Web with a refreshed framework in which information resources are described using logic-based knowledge representation languages. It aims to enable computers to automatically process information and to promote reusability and interoperability across heterogeneous systems. In recent years, the semantic web related technologies have been utilised to develop semantic-enhanced applications in various domains. Among these applications, semantic search supplements and improves conventional information retrieval systems on the basis of structural knowledge representation formalisms.

The paper provides a survey on existing semantic search systems, describes a prototype semantic search system, and provides discussion and future research directions in the semantic search research

Email addresses: eyx6ww@nottingham.edu.my, payam.barnaghi@nottingham.edu.my, andrzej.bargiela@nottingham.ac.uk (Wang Wei, Payam M. Barnaghi, Andrzej Bargiela).

area. The rest of the paper is organised as follows. Section 2 analyses some of the current semantic search frameworks and discusses their architectures and technologies. Section 3 describes the architecture of a semantic search system called IRIS. In section 4, we discuss our experiences in developing the system. Section 5 describes the future work and concludes the paper.

2. Related work

One of the drawbacks of the conventional information retrieval approaches is their limited capability of fulfilling user complex information needs, i.e., to find knowledge, especially on the web with tremendous amount of raw data in different format and under control of heterogeneous parties. Useful information or knowledge is buried in web pages which are designed primarily for human consumption. The lack of knowledge representation standards makes it difficult for software agents to perform logical reasoning to extract knowledge from multitude of information resources. Semantic-based search and retrieval approaches are considered as potential solutions to alleviate the situation. Using the semantic web technologies, resources are enriched with meta-data and encoded in ontologies using formal knowledge representation languages such as Resource Description Framework [1] (RDF) or Web Ontology Language [2] (OWL). Software agents are thus able to perform reasoning over the formally defined data to discover useful information or knowledge which is explicitly or implicitly represented by the resources. We select some of the existing semantic search systems to be discussed in this section.

2.1. *Simple HTML Ontology Extension (SHOE)*

SHOE is a knowledge representation language for internet applications [15]. Documents (i.e. web pages) are annotated with domain ontology using the SHOE language [14]. The SHOE search tool utilises a form-based interface and allows users to specify query context information by specifying property values for different categories in a selected ontology. Those values can be thought of as defining constraints rules for the retrieval process.

The design rationale behind the SHOE search system is that resources are annotated using a logic-based languages (i.e., SHOE). The search process translates user queries into constrained logical expressions and exploits the annotations to provide precise results. The main limitation is that the annotation is done manually which prevents it from being deployed in a large scale. Nevertheless, the work serves as a first step to the semantic search and outlines the process of a crisp logic based semantic search system.

2.2. *Large Scale Semantic Search (TAP)*

TAP is an infrastructure which provides a set of mechanisms for web sites to publish data and for applications to consume the semantic data on the semantic web [12,11,10]. It improves the traditional text search by understanding the denotation of the query terms and augmenting search results using a knowledge base with broad coverage. The infrastructure includes a number of components which are responsible for information extraction and integration, semantic annotation, semantic data publishing and semantic search [11,10]. The search module takes user queries and searches against the knowledge base to identify ontology terms that the queries detonate. To resolve ambiguities, a simple disambiguation step is provided with the user interface in an intuitive manner. TAP presents related information to users using an inference mechanism based on graph traversal. The infrastructure, although built for large scale applications, is based on a closed world assumption, i.e., mechanism for communication with ontologies and knowledge bases from other sources is not considered. Moreover, the nature of the knowledge base (i.e., general domains with broad and shallow coverage) and the search mechanism prevent more ad-hoc and fine-grained search applications from being designed.

2.3. *Question Answering (AquaLog)*

AquaLog is a question-answering system which accepts complex queries expressed in natural language and generates precise and meaningful answers inferred from its underlying knowledge base [19]. AquaLog employs a combination of techniques including natural language processing (i.e. GATE [7,8] framework), string matching algorithms, external

[1] http://www.w3.org/TR/rdf-schema/
[2] http://www.w3.org/TR/owl-guide/

lexicons such as WordNet [3], and a similarity service for relations and classes. Processing of queries and generating answers in AquaLog is analogous to a pipeline consisting of modules which translate queries expressed in natural language into query triples, match the query triples with ontology compatible triples, perform inference, and generate answers. AquaLog demonstrated its capability to answer some simple questions (e.g. which, what, who) by exploiting its knowledge base. A memory-based learning mechanism has also been embedded into the system which requires human intervention in situations where the system is not able to resolve query ambiguities.

2.4. *Image Retrieval (Falcon-S)*

Falcon-S [29] is an ontology-based semantic search engine for soccer images. It maintains a local image archive and a knowledge base about the soccer domain. The knowledge base is constructed by crawling the official soccer websites and parsing related web pages. The system has a term-object index constructed from the knowledge base as well as an object-image index allowing fast retrieval of images. Disambiguation of objects with same labels is done using context information derived from other terms in a query (if there are any). A novel feature of the system is that the ranking of results is enhanced using simple image processing techniques. By taking advantage of color feature of soccer jerseys and analysis of the color histogram, the system improves the image ranking by giving additional score to hits which have color records in the knowledge base [29].

2.5. *Semantic Search for Document Fragments (DOSE)*

DOSE is a modular multilingual architecture which integrates a number of interrelated components, namely, semantic mapping, annotation repository, semantic search, indexing, document fragment retrieval, and document substructure extraction [2,3]. One of the notable features of the system is that the annotation and search methods are based on document fragments. The intention is to provide users short text fragments which are likely to contain relevant information rather than document as a whole. The semantic search component accepts query terms and performs semantic mapping to obtain a weighted set of concepts. These concepts are then used to search annotation indices to fetch fragments and present the results to users or compose new documents. The tf-idf scheme [1] is used to measure and rank relevancy between document fragments and ontology concepts.

2.6. *Semantic Search for Multimedia Content (Squiggle)*

Squiggle is a semantic framework to help building domain-specific semantic search applications for indexing and retrieving multimedia items. Its underlying knowledge representation model is based on the SKOS [4] vocabulary [5]. SKOS is a model for expressing the basic structure and concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies, folksonomies as well as concept schemes embedded in glossaries and terminologies. In Squiggle, domain knowledge is not an integral part of the framework, as such, it architecture is domain-independent and can be instantiated to build domain-specific semantic search applications.

To implement a semantic search engine, Squiggle employs domain ontologies to empower searching, indexing and crawling processes. Query expansion is performed by a meaning suggestion module by exploiting the semantically interconnected terms in the domain ontology based on the SKOS vocabulary. Squiggle is demonstrated with two sample scenarios: Squiggle Ski image and Squiggle Music search. The limitation is that the searching and recommendation (i.e., query expansion) of multimedia information is mainly based on synonyms (e.g., people names in different languages).

2.7. *Knowledge and Information Management Platform (KIM)*

KIM introduces a holistic architecture of semantic annotation, indexing and retrieval of documents using an extensive semantic repository [16]. Similar to TAP, KIM platform consists of two main components: an upper level ontology which covers generic classes representing real world entities across various domains (such as People, Location, and Organi-

[3] http://wordnet.princeton.edu/

[4] http://www.w3.org/TR/swbp-skos-core-guide

zation) as well as their attributes and relations, and a heavily populated knowledge base which contains instances of the classes defined in the ontology. The automated annotation framework is based on the information extraction (IE) technologies [7,8] concerning named entities.

The semantic annotation of the content allows advanced semantic queries. For example, constrained queries with regard to entity type, name, attribute and relation can be formulated in order to obtain precise results as in [14]. Semantic annotations can also be used to match specific objects in documents to more general queries. For instance, a query such as "company, Redwood Shores" could be understood by the system using inference that the intention of the query is to retrieve documents mentioning the town and specific companies such as ORACLE and Symbian, but not the word "company" [16]. Although the proposed framework is plausible and promising, large scale experiments and user study need to be performed and compared with systems based on conventional information retrieval techniques to measure the effectiveness.

2.8. *Integrating Retrieval and Inference (OWLIR)*

In OWLIR [24,20] a document is represented as a combination of text, which is suitable for current web search engine's indexing and semantic markup. It adopts an integrated approach which combines logical inference and traditional information retrieval techniques. In OWLIR, the inference process is performed at three levels: documents indexing, query processing and results evaluating. A prototype system has been implemented to search and retrieve university event announcements. In the limited domain study, the authors report a significant improvements in precision compared to conventional full-text search engines.

2.9. *Relation-centered Semantic Search (SemSearch)*

SemSearch supports complex queries by providing comprehensive analysis of queries and translating them into formal queries which can be used for direct reasoning. Relation-centered search functionalities are provided by taking relations between user query terms into consideration [18]. The search process comprises four major steps: query processing to find out the semantic meanings of the keywords, translation of user query into formal query, searching the semantic data repositories, and finally ranking the results [18]. The formal query construction engine takes as input the matched semantic entities and outputs a set of formal logical query statements based on keywords combinations. For example, a semantic entity referred by a keyword can be matched to a subject, predicate or object in the knowledge base. Queries comprising two or more keywords are supported. However, if a query contains many keywords, the number of translated formal queries could be large and rules have to be introduced in order to reduce the complexity.

2.10. *Discussion*

We have identified some of the common characteristics among the semantic search frameworks discussed in the previous section. In most of the systems the semantic web technologies are integrated with existing technologies, ontologies or knowledge bases in respective domains have been developed; the implemented systems are mostly domain dependent unless the domain ontology is excluded from the underlying architecture [5] or the ontology is an upper level ontology like TAP [11] and KIM [16]; resources (e.g., documents or document fragments) are annotated with respect to corresponding ontologies to provide semantic representation of the original resources; most of the systems use simple reasoning methods to process relationships represented in ontologies and knowledge bases.

To process the user queries, AquaLog, DOSE and SemSearch [19,2,18] perform fine-grained query processing; OWLIR [24,20] supports inference at indexing, query processing, and result evaluating; DOSE and Squiggle [2,5] implement query expansion and refining; AquaLog [19] implements a memory-based learning mechanism to ensure the system's performance will evolve over time with users' feedback; DOSE [2] ranks the semantic annotations using the conventional information retrieval techniques. Table 1 provides a summary of the features of the studied systems.

In most of the studied work the inference is viewed as a simple graph traversal problem. It only exploits facts asserted in the knowledge base. Such inference is referred to as blind and shallow inference which might result in retrieval of irrelevant information [27] because entities retrieved from the graph which are "physically" connected to while are not necessar-

Table 1
Summary of the characteristics for the semantic search frameworks

Feature\System	SHOE	TAP	AquaLog	Falcon-S	DOSE	Squiggle	KIM	OWLIR	SemSearch
Ontology	Y	TAP	Y	soccer	Y	N	KIMO	Y	Y
Inference	shallow	shallow	Y	template	taxonomy	Y	Y	Y	Y
Query Processing	N	N	Y	N	Y	N	N	N	Y
Query Expansion or Refining	N	N	Y	N	Y	Y	N	N	Y
Relevance Feedback	N	N	Y	N	Y	Y	N	N	N
Ranking	-	-	-	Y	tf-idf	-	-	-	Y
RDF/OWL Repository	-	-	-	sesame	jena	sesame	sesame	sesame	sesame
String Matching	-	label	Y	N	tf-idf	N	N	N	Y
Indexing	N	Y	Y	Y	Y	Y	Y	Y	Y

ily semantically related to the original query. Some researchers present a different point of view and argue that power of the semantic-based systems is that even experienced users would find information they were not expecting [9]. Another point worthwhile to mention is that by introducing logical rules and logical reasoning mechanisms, semantic search systems could find previously implicit or even unexpected information or knowledge and consequently retrieve semantically related information. We will discuss logic rules and reasoning in more details in the next section.

3. The IRIS semantic search system

We have implemented a prototype of a semantic search system called IRIS (Information Retrieval In the Semantic web) which is dedicated to search scientific publications. In this section we present the architecture of IRIS and describe its components in detail. An initial evaluation is performed and the results are compared with three search engines, namely, the ACM [5] digital library, the Google Web Search [6], and the Google Scholar [7].

3.1. *Prevailing approaches for searching scientific literature*

Researchers often start to explore published literatures either by using web search engines (i.e. Google Scholar) and search engines provided by large digital libraries (i.e. ACM, IEEE X*plore* [8]), or by following citation links in currently referred articles. ACM states that "the relevance of a document is based on how many of the search terms are present in the document, how frequently the search terms occur, and how close the search terms are to each other". IEEE X*plore* introduces its search strategy as "search results are weighted and sorted by relevance based on an algorithm using the following parameters: frequency of the search term in the full abstract/citation record; text length of the abstract/citation record". From the statements one can see that both of the search systems resemble conventional keyword-based search approach. It is worthwhile to note that they both retrieve citations which allows users to explore related papers published previously in similar areas.

Another prevailing approach for searching scholarly literature is the citation indexing that is based on the link analysis, for example CiteSeer [9] and Google Scholar. CiteSeer's Autonomous Citation Indexing (ACI) helps organising the literature by automating the construction of citation indices. It is able to promote the visibility and dissemination of more literature compared to manually maintained citation index [17]. Documents or citations returned by the citation indexing can be ranked based on the number of citations made to them. The citation indexing provides effective ways for browsing literature through the citation links. However, it also has some limitations such as the assumption that a large number of citations imply scholarly impact is not always true [17]; citation statistics for recent publications may not be available because that it takes some time for those to be referenced by others. Google Scholar searches and ranks scientific articles based on a combination of parameters such as weighting the full text, the author, the publication in which the articles appears, and how often the articles have been cited in other scholarly literatures [10]. The utilization of the links between citations and

[5] http://portal.acm.org/
[6] http://www.google.com/
[7] http://scholar.google.com/
[8] http://ieeexplore.ieee.org/
[9] http://citeseer.ist.psu.edu/
[10] http://scholar.google.com.my/intl/en/scholar/about.html

contents as well as a semantic-enhanced approach is used in our current research to design IRIS.

3.2. *The IRIS Semantic Search System*

The IRIS semantic search system is developed to complement current approaches towards searching and exploring of scientific publication repositories. The architecture of the system is illustrated in Figure 1.

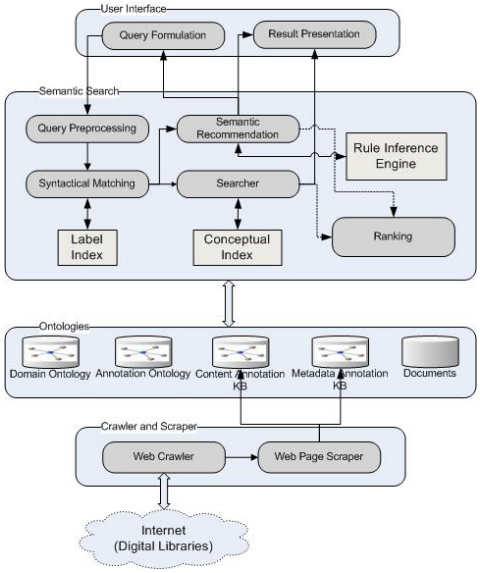

Fig. 1. The architecture of the IRIS semantic search system

The web crawler retrieves relevant web pages from the Web (mainly large digital libraries) and a web page scraper (based on the NekoHtml API [11]) parses, analyzes, and extracts needed data from the crawled pages. The ontology (knowledge representation) layer comprises of a set of ontologies, i.e. domain ontology, annotation ontology, knowledge bases, and the content annotation for the documents. The semantic search layer can be abstracted into eight modules: query preprocessing, syntactical matching, label index, conceptual index, semantic recommendation, searcher, rule inference engine and ranking. In the following sub-sections we elaborate individual modules of the system.

[11] http://www.apache.org/ andyc/neko/doc/html/

3.2.1. *IRIS Ontologies*

The design principle of the IRIS ontologies [12] is to reuse existing standards, leverage the expressivity and complexity of existing ontologies, and only define additional vocabularies where necessary. We select elements from set of common vocabularies such as, Dublin Core [13] (DC), Friend-Of-A-Friend [14] (FOAF), SKOS, and UBMC [15] publication ontologies. The resulting IRIS annotation ontology comprises classes and predicates drawn from the four aforementioned ontologies as well as some elements (e.g., class and property) defined for the project. Doing so provides an interface which is open to the existing semantic web world. data integration and reuse with external data sources can be realized without much overhead of data consolidation and schema mapping. Figure 2 shows part of the annotation ontology.

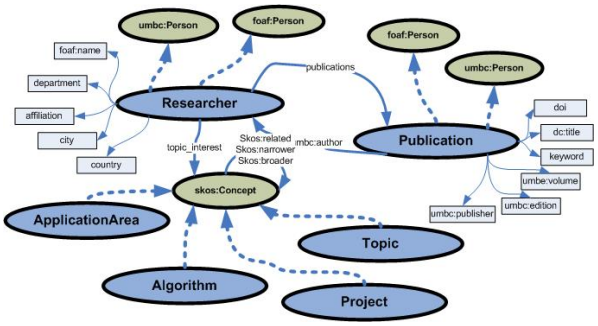

Fig. 2. The IRIS annotation ontology.

The IRIS domain ontology consists of concepts in the computer science domain, particularly in machine learning area. The ontology can be viewed as a comprehensive extension of the ACM Classification Tree [16]. The tree is a bit out-dated and is not sufficient for our design purpose. For example, one cannot find research topics such as machine leaning and semantic web in the hierarchy. We convert the OWL version of the tree [17] into RDF by converting classes into instances with relations drawn from the SKOS vocabulary due to the fact that it is difficult to instantiate semantic relations between classes. We add an extra

[12] http://rosie.nottingham.edu.my:8080/iris-v0.1/ontology.jsp
[13] http://purl.org/dc/elements/1.1/
[14] http://xmlns.com/foaf/0.1/
[15] http://ebiquity.umbc.edu/ontology/
[16] http://www.acm.org/class/1998/
[17] http://cse.unl.edu/ scotth/SWont/acmCCS.owl

child node "Machine Leaning" under the node "I.2_ARTIFICIAL_INTELLIGENCE\I.2.6_Learning" and extend the node with sub-topics in the "Machine Learning" such as "Supervised Learning", "Concept Learning", "Instance-based Learning", and so on. We also define some other classes and their instances, for example, "Concept" (i.e., Hypothesis Testing), "Algorithm" (i.e., Back Propagation), "Application Area" (i.e., Pattern Recognition). The instances are interrelated using predicates defined in the SKOS vocabulary.

3.2.2. *Document Annotation*

The IRIS utilizes a document annotation differing from existing approaches. Most of the existing systems utilize information extraction techniques such as KIM [16] which is built on top of the GATE [8] framework; DOSE [2] uses a simple processing module which takes an ontology and returns a set of ontology terms for each input resource using string matching; in a different approach TAP [12] annotates documents by parsing documents using predefined templates.

We use a conventional search engine to query document repositories recursively using the ontology terms. The returned documents are then associated with the query terms with which they were located. An issue has been considered is that the frequency of the concepts is not preserved as most of the semantic search approaches do. This raises the consideration that documents might be annotated with out-of-context terms that are occasionally mentioned by authors. To a certain extent, it can be alleviated using the approach proposed in our previous work by analyzing the document annotations with document abstract and eliminating concepts which are far away from the main topic [27]. The intuition is that there are just one or few topics per document.

3.2.3. *Inference Process*

In most of the existing work the inference process is modeled as the graph traversal problem. In IRIS the inference is not only based on assertions in the knowledge bases but also predefined rules. Once a user submits a query to the semantic search engine, the engine will search a label index and find the closest matching ontology concepts. Upon searching for documents the matched ontology concepts are sent to a Prolog engine (we use SWI-Prolog [18]) to obtain conceptually related, broader, or narrower concepts using the predefined rules. The process is done by the semantic recommender module in the system. After inference with the rules, more relevant concepts are found and presented to users. The inferred concepts can be used either for query expansion and refinement, or to help users navigate and explore the relevant publications.

3.2.4. *Ontology and Document Indexing*

In order to enhance query processing the system implements two indices (we use Apache Lucene [19]). The first index, the label index, is used for syntactical matching, a process that finds corresponding ontology concepts which match the query terms based on string syntax similarity measure. The similarity measure uses the edit distance algorithm [20]. The document index, or conceptual index currently indexed about 50,000 scholarly articles in the machine learning area. The most important information stored in the document index is the ontology concepts annotating content of the documents. An ontology concept is different from a keyword because it has explicitly defined meaning and relations to other concepts. Other information such as authors, publication date, proceeding, and abstract is also saved into the index. The information stored can be used to extend the functionalities offered by the current implementation of the IRIS, for example, to develop entity-centric search services where users are interested in finding more information about a particular entity. Moreover, the abstract can be used to construct a conventional keyword-based search index to be integrated within the semantic search framework.

3.2.5. *Query Processing*

The classical IR systems have some standard query processing steps such as accent and spacing detection, stop words elimination and stemming for both documents and queries [1]. It is normally assumed that the user queries do not consist of "variables" [26], for example, a query for "people" and a specific "company" in a keyword-based retrieval system does not really return the actual people who have relations with that company, instead the query will probably return documents in which the word "company" and the "people" co-occur. In contrast,

[18] http://www.swi-prolog.org/

[19] http://lucene.apache.org/

[20] http://www.nist.gov/dads/HTML/Levenshtein.html

in a semantic search system, such a query will be interpreted as instances of the "People" class who has various relations with the "Company", for example, the relations could be "work-for" or "has-business-relation-with".

In the IRIS system, the query processing module scans the query terms and tries to identifies the classes and instances. If there is no class involved, the search engine will look for documents in which the concepts appear. At the same time, it will query the knowledge base to retrieve semantically related concepts. This process offers some advantages, for instance, the system is able to help the users to resolve ambiguity (i.e., concepts having the same label while they refer to different meanings under different context; for example, the concept "online training" in the context of machine learning and in the context of computers and organizations); if the user's vocabulary is different from the one used in the ontology, the system is able to show the vocabulary used in the system. In some circumstances that a user might not have a clear idea about what to search initially, the system is able to help generating more effective query terms using the concept recommendation or navigating through the concept graph. If IRIS observes that there is any variable(s) or class(es) appearing in the query, the query will be interpreted as finding all objects that are instances of that class. On one hand, the system retrieves documents that are pertaining to the identified concepts, on the other hand, the semantic recommender module presents semantically related, broader and narrower concepts. An effective way which allows users to specify more variables and search for more specific information is to provide an interface for advanced search. With the advance interface users are able to construct more constrained queries using relations between objects to retrieve more precise and relevant results.

3.2.6. *Search Process*

The user's original query is submitted to the query parser for preprocessing (e.g. tokenizing, eliminating stop words). Then the processed query keywords are used to search the label index through the syntactical matcher module to obtain matching ontology concepts. An additional disambiguation step is added when the string similarity measure between query terms and ontology terms is below pre-defined threshold. The set of ontology concepts are then forwarded to the query interpreter for further processing. The interpreted concepts are sent to the concept recommender for inference using the Prolog engine and rules to generate semantically related concepts as recommendations, and at the same time the conceptual index is searched to retrieve matching documents. Currently the ranking of the documents is based on the Apache Lucene results. The ranking mechanism for the semantic search is further discussed in section 4.

3.3. *Initial Evaluation*

For the current implementation of the IRIS system, we adopt a different evaluation approach instead of the classical methods such as recall, precision and F1 measures. The approach is to evaluate the effectiveness of the search engine based on the semantic recommendations. The ranking function is not considered at this stage. The evaluation method is outlined as follows:

- Generating query terms and submitting them to IRIS, the ACM digital library, Google (restricting the domain to the ACM), and Google Scholar respectively;
- Evaluating the number of returned hits, number of irrelevant hits of the four search engines;
- Evaluating the concepts returned by the IRIS search engine and checking the appearance of these concepts or topics in documents returned by the ACM and Google.

The queries and results are presented in Table 2. From the table, M1 denotes number of returned documents; M2 denotes number of irrelevant results in the top 40 results; and M3 is the number of related concepts suggested by the search engine. Since Google and ACM do not provide similar results, M3 measure is only applicable for the IRIS system. Fifteen queries were submitted to the search engines. Relevance of the top 40 documents was checked by reading their abstract or text snippets. If the abstract or the snippet is not sufficient for determining the relevance, full text is used to make judgment. The relevance judgment is subjective, and the evaluators were told to mark the document as relevant if they regard those documents as potential candidates for references assuming they are writing papers in the related areas.

The four search engines in fact are not comparable directly because the document collections under evaluation vary in scope. The Google Scholar indexes research papers or citations crawled from

Table 2
Query results from four search engines

Query\Search Engine	IRIS			ACM			Google Web			Google Scholar		
	M1	M2	M3	M1	M2	M3	M1	M2	M3	M1	M2	M3
supervised learning	1184	4	58	1293	4	-	8400	0	-	43800	0	-
halving algorithm	23	8	2	11	6	-	16	2	-	315	14	-
weighted majority algorithm	89	5	2	172	4	-	556	6	-	641	12	-
reinforcement learning	536	3	3	2818	1	-	6500	0	-	38600	1	-
temporal discounting	3	0	1	6	0	-	8	0	-	10	0	-
artificial neural networks	4407	10	17	6373	2	-	111000	0	-	165000	0	-
spreading activation	319	3	1	450	2	-	1570	0	-	11500	0	-
quadratic classifier	2	0	2	38	2	-	31	0	-	872	4	-
data mining	2424	1	30	20013	1	-	71200	0	-	33500	0	-
concept learning	651	20	7	722	16	-	6770	1	-	18300	7	-
naive Bayes classifier	560	0	4	328	0	-	849	0	-	3000	0	-
decision tree learning	0	0	7	431	0	-	974	0	-	3860	3	-
medical diagnosis, machine learning	0	0	4	1412	11	-	5910	2	-	59100	0	-
concept learning, algorithm	651	10	4	19217	20	-	17000	11	-	271000	26	-
machine learning, application area	0	0	16	16022	7	-	12500	0	-	52400	1	-

the Web, while the IRIS mainly crawls information from the ACM Digital Library. Although we limit the Google Web search to the ACM domain, it retrieves citations from the ACM guide (some publications belong to other parties). Moreover, the ranking of the four search engines vary significantly. From the figures one can see that the Google Scholar and Google Web achieve better results due to their broad indices (More than half of the Google Web search results are citations). Although currently there is no advanced semantic ranking function in the IRIS, the figures of the top forty results show comparable performance with the ACM Digital Library search engine. This implies that our simple annotation approach works effectively. One of the notable features of IRIS is that it always recommends conceptually related concepts (i.e. research topic, algorithm, application area, etc) immediately after processing user queries. On the contrary, the other three search engines do not provide such recommendations explicitly. Only small portion of the concepts (less than 10% of the results) recommended by IRIS can be observed in others' hits. The recommended concepts help users find out what are the sub-topics, related concepts, popular algorithms, successfully applied application areas, etc. This is particularly effective when user queries are broad (e.g. supervised learning). Consequently, the system enables researchers to identify topics of research publications which could be a good starting point in order to explore more related research articles.

4. Discussion

A semantic search approach improves the conventional search methods from a different perspective by looking at the meaning of the words. Consequently, it extends the scope of traditional information retrieval techniques from document retrieval to entity retrieval [23,9,13]. In terms of document retrieval, which is a specific entity, semantic-based methods can be used to build a metadata based search engine which does not require access to full text of the publications, and consequently requires much less computation. More importantly, in semantic search systems, content of documents is represented using knowledge representation formalism which facilitates automatic reasoning. For example, given a document pertaining to a topic, the search engine is able to retrieve documents whose topics are related to, broader, or narrower than the original topic. The results can also be filerted based on entity type, for example, documents whose main topics are instances of "Algorithm" or "ApplicationArea" class. Moreover, the document collection can be organised with respect to the domain ontology to facilitate user browsing.

The document annotation is represented using machine-processible knowledge representation language such as RDF or OWL, it is ready to be used by other software agents in different application settings. Despite the simiplicity of our annotation approach, the system shows that such representation is sufficient and effective for document retrieval purpose. The intuition is that more terms appearing in

the article which match query terms does not necessarily indicate its higher relevancy or quality. Detailed annotation might be able to provide more accurate approximation of the document representation; however, due to the fact that scientific terms (nouns or noun-phrases) are apparently more important than common verbs, adjectives, adverbs, and so on, it is not clear how much natural language processing techniques can contribute to the annotation and search process in this application setting.

The domain ontology used in IRIS helps to alleviate the synonym problem. Different labels of the same concept used frequently by different people have been encoded in the domain ontology. For example, the term "support vector machine", "SVM", and "large margin classifier" in fact denote the same concept. In scientific research, polysemy does not happen as often as other general areas. The occurrence of polysemy can only be solved by considering the context of term where it is appeared. For example, "online training" in machine learning and in general computer literatures carry different meanings. The semantic search framework also provides concepts formalization using ontologies. In research articles scientists often use some of the terms interchangeably such as concept, category, topic, research area and so on. In the IRIS ontology we have defined classes such as "ApplicationArea", "Algorithm", "ResearchArea", "MathematicalConcept", "Model", "Framework", and so on. This allows the system interpret the query "machine learning, applications" in a meaningful way and retrieve articles which describe applying machine learning techniques into some application areas such as text categorization or bioinformatics. On the contrary, a conventional search engine in some of the cases will find those articles in which the three words co-occur. Neither will a citation indexing based search engine would be able to interpret the meaning of the query.

One of the most prominent features of the semantic search approach is to suggest semantically related concepts based on logical inference to help users navigate in the concept graph or formulate more precise queries. The searcher does not necessarily need to know the underlying knowledge base and vocabulary well. The inference could be done using either a logical deductive approach or inductive methods such as by mining association rules. The suggested concepts can also be ranked based on a probability model such as the random walk model.

Finally, devising an appropriate ranking mechanism is a challenging task for documents retrieved by semantic search systems due to lack of features or data (i.e., there are not as many ontological concepts as words in a document). A conservative approach would be using existing ranking methods such as citation-based ranking [17] by analyzing and counting citation links between publications. Recently there have been some researches to design intelligent search applications using the idea of collaborative filtering based on communities or social networks analysis to support ranking and measuring the quality of search results [25,22]. The underlying assumption is that users from the same community are likely to have similar information needs. Documents viewed by members of the community are likely to represent the community's interest. By exploiting the collaborative behaviors of the community members and by recording and analyzing user requests and server logs, information that has been chosen by many members frequently in the past is recommended by the system in response to similar requests. However, treating members in a community or network as equal, the real power of the social network analysis is not properly exploited. For example, the expertise of an actor, the positions she has, and her influence on others actors (e.g. degree, closeness and betweenness centrality measures [28,21]) are different and can be used as an important parameter for ranking documents. On the other hand, limitations of the citation-based ranking has been identified by some researchers such as the time delay to establish citation links, and the context and purpose of the citations [17,22] (i.e., commendation or criticism). The two paradigms could complement each other and measure ranking values based on combination of the citation-based and social network-based metrics which exploits advantages of the both. For example, the newly-published work by an expertise can appear at a higher ranking before its citation links are established.

5. Future Work and Conclusion

The observation that conventional search techniques function ineffectively in situations where finding knowledge is more substantial has motivated the research of enhanced search paradigms. Semantic search is an effort to exploit meaning of data and automated logical reasoning to retrieve information in a meaningful and precise manner. We have investigated related research in the past few years. The studied pilot semantic search systems

are presented in chronological order to show how the related research has evolved over time. Commonalities, distinctiveness, strengths, and weaknesses of the systems are compared and discussed. We demonstrate the IRIS semantic search engine developed in our research lab. The architecture and individual modules of the system are elaborated. The initial evaluation shows some notable features and results compared to other search systems, although the search result does not surpass existing search engines due to narrow coverage of document collections and the lack of an advanced ranking mechanism.

The experiences of developing the semantic search framework suggest that the power of the semantic search lies in finding knowledge which is extremely difficult if not impossible to obtain using current search techniques. The current IRIS focuses on providing a complementary approach for scholars to explore tremendous number of scientific publications. We observe that researchers might be interested in finding other research related information, for example, who are the involved in various research areas or communities? Which universities or institutions are prominent in those research areas? How researchers are related to each other? Answers to these questions are valuable to different group of users, for instance, a postgraduate student might be interested in identifying active researchers or institutions in certain research area; funding agencies might need this sort of knowledge to decide on distributing their funds [23]; a research institution might want to find other groups for research collaboration. To answer these questions using current web search engines users have to go through a number of information sources in order to derive answers. The semantic search mechanisms facilitate to derive knowledge and answer the user queries by interpreting the meaningful relationships between information resources. The future work will concentrate on knowledge mining and developing search services which support answering the above mentioned questions.

A tedious process in developing the IRIS system is that the domain ontology is developed manually. The major problems of the manual construction are cost of development, limited scope, and difficult maintenance. There are various algorithms and approaches to automatically derive knowledge and construct ontology from domain-specific text collections including concept hierarchy induction, learning attributes and relations, and ontology population [6]. The plausible assumption is that sufficient amounts of texts provide a reasonable coverage of the domain knowledge. The future work will also involve domain ontology construction utilizing automatic approaches in various research areas from large amount of documents.

References

[1] R. A. Baeza-Yates, B. A. Ribeiro-Neto, Modern Information Retrieval, ACM Press / Addison-Wesley, 1999.

[2] D. Bonino, F. Corno, L. Farinetti, Dose: A distributed open semantic elaboration platform, in: ICTAI, IEEE Computer Society, 2003.

[3] D. Bonino, F. Corno, L. Farinetti, A. Bosca, Ontology driven semantic search, WSEAS Transaction on Information Science and Application 1 (6) (2004) 1597–1605.

[4] T. Burners-Lee, J. Hendler, O. Lassila, The semantic web, Scientific American 284 (5).

[5] I. Celino, E. D. Valle, D. Cerzza, A. Turati, Squiggle: a semantic search engine for indexing and retrieval of multimedia content, in: Proceedings of the 1st International Workshop on Semantic-Enhanced Multimedia Presentation Systems (SAMT 2006), 2006.

[6] P. Cimiano, Ontology Learning and Population from Text: Algorithms, Evaluation and Applications, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[7] H. Cunningham, Information Extraction, Automatic, Encyclopedia of Language and Linguistics, 2nd Edition.

[8] H. Cunningham, K. Bontcheva, Computational Language Systems, Architectures, Encyclopedia of Language and Linguistics, 2nd Edition.

[9] H. Glaser, I. C. Millard, Rkb explorer: Application and infrastructure, in: Proceedings of Semantic Web Challenge, 2007.

[10] R. V. Guha, R. McCool, Tap: a semantic web platform, Computer Networks 42 (5) (2003) 557–577.

[11] R. V. Guha, R. McCool, Tap: A semantic web test-bed, J. Web Sem. 1 (1) (2003) 81–87.

[12] R. V. Guha, R. McCool, E. Miller, Semantic search, in: WWW, 2003.

[13] A. Harth, A. Hogan, R. Delbru, J. Umbrich, S. ORiain, S. Decker, Swse: Answers before links!, in: Proceedings of Semantic Web Challenge, 2007.

[14] J. Heflin, J. Hendler, Searching the web with SHOE, in: Artificial Intelligence for Web Search. Papers from the AAAI Workshop. WS-00-01., AAAI Press, Menlo Park, CA, 2000.

[15] J. Heflin, J. Hendler, S. Luke, SHOE : A Knowledge Representation Language for Internet Applications, Tech. rep., Department of Computer Scienceand University of Maryland at College Park (1999).

[16] A. Kiryakov, B. Popov, I. Terziev, D. Manov, D. Ognyanoff, Semantic annotation, indexing, and retrieval, J. Web Semantics. 2 (1) (2004) 49–79.

[17] S. Lawrence, C. L. Giles, K. Bollacker, Digital libraries and Autonomous Citation Indexing, IEEE Computer 32 (6) (1999) 67–71.
[18] Y. Lei, V. S. Uren, E. Motta, Semsearch: A search engine for the semantic web, in: S. Staab, V. Svátek (eds.), EKAW, vol. 4248 of Lecture Notes in Computer Science, Springer, 2006.
[19] V. Lopez, M. Pasin, E. Motta, Aqualog: An ontology-portable question answering system for the semantic web, in: A. Gómez-Pérez, J. Euzenat (eds.), ESWC, vol. 3532 of Lecture Notes in Computer Science, Springer, 2005.
[20] J. Mayfield, T. Finin, Information retrieval on the semantic web: Integrating inference and retrieval, in: Proceedings of Workshop Semantic Web at the 26th Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, 2003.
[21] P. Mika, Flink: Semantic web technology for the extraction and analysis of social networks, J. Web Semantics 3 (2-3) (2005) 211–223.
[22] A. Mislove, K. P. Gummadi, P. Druschel, Exploiting social networks for internet search, in: Proceedings of the 5th Workshop on Hot Topics in Networks (HotNets'06), 2006.
[23] N. Shadbolt, N. Gibbins, H. Glaser, S. Harris, M. M. C. Schraefel, Cs aktive space, or how we learned to stop worrying and love the semantic web, IEEE Intelligent Systems 19 (3) (2004) 41–47.
[24] U. Shah, T. W. Finin, A. Joshi, Information retrieval on the semantic web, in: CIKM, ACM, 2002.
[25] B. Smyth, A community-based approach to personalizing web search, Computer 40 (8) (2007) 42–50.
[26] C. J. van Rijsbergen, A new theoretical framework for information retrieval, in: SIGIR, ACM, 1986.
[27] W. Wang, P. M. Barnaghi, A. Bargiela, Semantic-enhanced information search and retrieval, in: Proceedings of the 6th International Conference on Advanced Language Processing and Web Information Technology, 2007.
[28] S. Wasserman, K. Faust, Social network analysis: methods and applications, 1st ed., Cambridge Univ. Press, Cambridge, 1997.
[29] H. H. Wu, G. Cheng, Y. Z. Qu, Falcon-s: A ontology-based approach to searching objects and images in the soccer domain, in: ISWC, 2006.