A Survey of Semantic Keyword Search Approaches

Siraya Sitthisarn
Department of Computer Science and Information Technology
Thaksin University
Patthalung, Thailand
e-mail: ssitthisarn@gmail.com

Lydia Lau and Peter M Dew
School of Computing
University of Leeds
Leeds, UK
e-mail: {L.M.S.Lau,P.M.Dew}@leeds.ac.uk

Abstract—**A broad range of approaches to semantic information retrieval has been developed in the context of semantic web concept. Ontologies take the significant role for improving the precision of search. In the last few years, there has been an increase in the amount of information stored in semantically enriched knowledge bases, represented in RDF format. To access RDF information, the semantically formal queries are required. However framing such queries is inappropriate for inexperience users because they need specialist knowledge of the underlying ontology and syntax. Therefore an easy-to-use interface is required. Many semantic searches provide solutions to solve this limitation. This survey introduces interfaces of semantic search systems built on the top of semantically enriched knowledge bases. Semantic keyword search systems are particularly focused. The common idea of semantic keyword search architecture is presented and six semantic keyword search systems implemented by different approaches were investigated. We briefly discuss comparison of them by our criteria. The criteria were derived from the common ideas and technical implementations used in these approaches. The evaluation methods in three features: effectiveness, efficiency and usability are also explored to guide researchers on how to conduct experiment evaluations. Finally, the open issues on semantic keyword searches are raised. These give directions for future application development and the research.**

Keywords- semantic keyword search; semantic web; semantic search interface

I. INTRODUCTION

Semantic web is intended for including meaning to information and representing it in a format that can be understood by applications or agents. It provides the necessary infrastructures for publishing and determining ontological descriptions of concepts. This improves human and computer collaboration on the internet. One important issue of semantic web focuses on the information management, particularly, the semantically supported information retrieval called semantic search.

Semantic web provides a number of technologies. Ontologies take a significant role for improving the accuracy of web and information searches [1]. This is because the semantic search is referred to a precise concept in ontologies, instead of keyword's terms in documents that are generally ambiguous. Another main technology for semantic search is RDF (Resource Description Framework). RDF is a formalized language for representing information in the web. It is based on XML-based syntax. RDF enables computer applications/agents to understand content of information. Also, the information exchange between applications can be performed without the problem of syntax and loss of meaning.

In the last few years, there has been an increase in the amount of information stored in semantically enriched knowledge bases, represented in RDF format. Also, the concept of Linked data has been widely acceptable to a researcher community [10-12]. This concept refers to a set of practices for publishing and connecting RDF data on the web. It leads to the creation of a global data space from connecting diverse data sources. This will become a main search space in the future.

Recently, a number of ontology-driven semantic search approaches have been reported [2-9]. Their application domains and their realisation are different. Some systems dealt with RDF information. However, there is a challenge to query/access the RDF information, it requires semantic queries but framing such queries is inappropriate for most potential users. Since, it needs specialist knowledge of the underlying ontology and syntax of the query language [13]. Therefore, an easy-to-use interface is required. To solve the above problem, many researchers have explored systems based on a semantic keyword search approach [3, 4, 7, 14-16]. The keyword interface is interested because it provides a familiar interface similar to one adopted by traditional web searches (e.g. Google) for non-specialists users.

In this paper presents the common idea of semantic keyword search architecture. Six semantic keyword search systems implemented by different approaches were surveyed and analysed. Comparison criteria are proposed to compare these approaches. This work will be a first step to build an understanding of current ideas, architecture, and technical implementations and algorithms used in semantic keyword search systems. In addition, evaluation methods were investigated to guide researcher on how to conduct effectiveness, efficiency and usability evaluations. With regard to our survey, open issues are raised. These offer directions for researchers and application developers to improve the approaches further.

The paper is structured in the following way. The next section explains role of ontologies and RDF to improve information retrieval. Section III gives detail of fundamental concepts in semantic search. This explains the definition of a semantic search and its main characteristic categorised by type of information. Section IV discusses categories of interfaces that semantic search systems provide for entering a search query. Then the common idea of semantic keyword search system architecture is proposed in Section V. This shows the main components of the systems. Section VI reviews six semantic keyword search systems and the comparison of these systems is proposed in Section VII. The survey of evaluation methodology is presented in Section VIII. Section VIIII provides the open issues which are needed to be solved by a research community.

II. Role of Ontologies and RDF to improve information retrieval

The main challenge of search engines is to provide high retrieval effectiveness. However, the precision of results that traditional keyword search engines provide is still not satisfied. This is because the search engines have neither understanding of the context of the keyword query nor content of information. Semantic web, in particular, ontologies and RDF technology, enables solutions for these problems.

A. Ontology

The definition of ontology was proposed originally by Gruber [29], when he asserted that "ontology is an explicit specification of conceptualization". It is used for formally represented knowledge based on a conceptualization of "the object, concepts, and other entities that are presumed to exist in some area of interest and the relationship that holds among them" [30]. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose.

An ontology is used to model data at the semantic level. Significantly, an ontology is aimed at a shared understanding of terminologies in domains, which is necessary to overcome differences in terminologies between heterogeneous sources/applications [31, 32]. As depicted in Fig. 1, the ontology is to formally describe the vocabulary related to expert witness information. The expert witness information ontology consists of main concepts and relationships that directly relate to expert witness information, such as concept of "ExpertWitness", "ExpertiseArea", "Dispute Case" and "DisputeSubject".

An ontology is used as a schema for representing information, which allows search engines to understand the meaning of the information. In addition, an ontology enables improving the precision of searches. The search engines can look for information that refers to a precise concept corresponding to user keywords instead of collecting all information in which certain, generally ambiguous keywords occur. In this way, difference in terminologies between the information and the queries can be overcome.

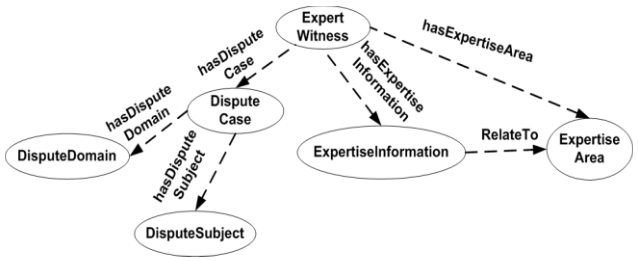

Figure 1. Ontology of expert witness information

B. RDF

RDF (Resource Description Framework) is a formalized language for representing information in the web [28]. It is aimed at representing information which needs to be processed by an application rather than only being displayed to people.

RDF provides a common set of assertions, know as statements, for expressing information. Each statement consists of three elements: subject, predicate and object. The three elements of statements have meanings that are analogous to their meanings in English grammar. The subject is the thing that statement describes. The object is property of subject, while the predicate is the relationship between subject and object.

RDF is a graphical representation in which statements form a directed graph. Subject and object are represented as nodes and the predicates are represented as edges. There are two types of nodes: resources and literals. Literals represent a constant value, such as a number or a string. In contrast, resources representing everything else by using URI and resources can be either subjects or objects. Predicates represent the connection between resources, or between resource and literal.

Fig. 2 is an example of RDF graph, showing a representation for an expert witness named Andrew Burton. As we create an RDF graph of nodes and edges, a URI reference used as a graph node identifies subject (*http://foaf/Andrew.foaf*) and object (*http://www.Andrew Description*). A URI used as a predicate identifies a relationship between the things identified by the connected nodes. In this graph, objects are also represented by literals such as *literal: Andrew.*

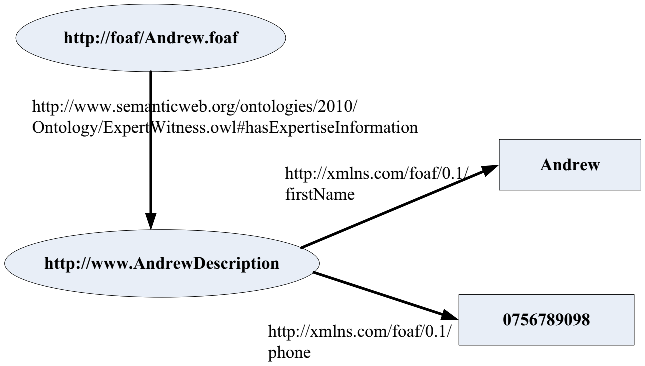
http://foaf/Andrew.foaf
http://www.semanticweb.org/ontologies/2010/Ontology/ExpertWitness.owl#hasExpertiseInformation
http://xmlns.com/foaf/0.1/firstName
Andrew
http://www.AndrewDescription
http://xmlns.com/foaf/0.1/phone
0756789098

Figure 2. An RDF graph representing expert witness information named Andrew Burton

RDF also provides an XML-based syntax called RDF/XML for recording and exchanging graphs. Fig. 3 shows a small chunk of RDF information in RDF/XML corresponding to the graph in Fig. 2 and this example of RDF information is associated to expert witness information ontology in Fig.1.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:j.2="http://www.semanticweb.org/ontologies/2010/
Ontology/ExpertWitness.owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs=http://www.w3.org/2000/01/rdf-schema#>

<j.2:ExpertWitness rdf:about="http://foaf/Andrew.foaf">
  <j.2:hasExpertiseInformation>
  <j.2:PersonalDescription
rdf:about="http://www.AndrewDescription">
   <foaf:firstName>Andrew</foaf:firstName>
   <foaf:surname>Burton</foaf:surname>
   <foaf:e_mail>andrew@hotmail.com</foaf:e_mail>
   <foaf:phone>07567890987</foaf:phone>
  </j.2:PersonalDescription>
  </j.2:hasExpertiseInformation>
</j.2:ExpertWitness>
</rdf:RDF>
```

Figure 3. An example of RDF information

Information representation using statement is a powerful tool for information retrieval. Since search engine can understand meaning of content and return more relevant result to users. In addition, RDF enables information exchange without the problem of syntax and loss of meaning.

III. FUNDAMENTAL CONCEPTS IN SEMANTIC SEARCH

Semantic search uses semantic web technologies to provide an improved form of search where meaning and structure are added to the content of information. These are then used for searching and extraction of answers for user's query [17]. Currently, a number of semantic search systems use ontologies to clarify user's intention and to expand user's queries [3-5, 7, 18, 19]. An ontology-driven semantic search can be characterised either as built on top of a semantic knowledge base or built on top of a vector space machine (i.e. a conventional search engine)[18]. The information in a semantic knowledge base is represented in RDF format. While a vector space machine deals with information in text file. The semantic search built on a knowledge base aims to answer questions by browsing ontologies/RDF information, while the latter one is focused on improving large-scale search results.

A. Semantic search built on knowledge base

The systems based on these approaches use reasoning mechanism and ontology querying languages to access and retrieve RDF instances from a knowledge base [3, 4, 7]. Therefore, these approaches are focused on retrieving instances associated to URI of information rather than documents. A semantic formal query such as SPARQL is used to access a knowledge base. Nevertheless, inexperienced users may suffer from lack of background knowledge on the ontology structure and formal query syntax.

With regard to Fig.4, a example of SPARQL for access the RDF information is shown in Fig.3.

```
PPEFIX foaf:http://xmlns.com/foaf/0.1/
SELECT ?x
WHERE {?x foaf:firstName ?name.}
```

Figure 4. An example of a SPARQL query

This query retrieves all statement patterns where the property is 'foaf:firstName' and the object can be anything. In fact, when this query is executed, it will retrieve all resources of people who have 'name'.

B. Semantic search built on vector space mechine

These approaches combine semantic search with a traditional vector space model. Some start with semantic querying using semantic formal query languages (e.g. SPARQL, RDQL, OWL-QL) and use resulting instances to retrieve relevant documents using the vector space model [19, 20].

IV. APPROACHES TO CAPTURE AND PROCESS SEARCH QUERIES

In regard to knowledge base approach, we can identify 4 approaches of semantic search systems according to the user interface they provide for entering a search query [14].

A. Form based search

These systems provide web forms that allow users to specify a query associated with a concept, property or values in a semantic knowledge base. The Shoe search engine [9] is an example of the form based search engine. This form is suitable for users who are familiar with the concepts in the back-end ontology. The form based search engine is easy to implement but it is not flexible for users to use their own vocabulary for formulating a query.

B. RDF-based querying language fronted search

These systems rely on users entering a RDF-based query language to conduct the search. The Corese search engine [5] is an example in this category. Such search engine usually provides a sophisticated query language to support semantic data queries. However, the main limitation of this search engine category is that end users need to master the back-end ontology structure and the complexity of the semantic query language syntax.

C. Semantic keyword search

This semantic search approach enhances the performance of keyword search technique by transforming keywords into a semantic query automatically. The benefits of semantic keyword search systems are that they provide an easy search interface that users are familiar with by hiding the ontology structure and the complexity of the formal semantic query from users. However, the challenge of this approach is in the automatic construction of the formal semantic query which is relevant to user's query intention, represented by the keywords entered.

D. Natural language interface

A natural language query is an input for this approach. For example, the FREyA system [2] uses natural language processing technologies to reformulate a natural language query into a SPARQL query. It transforms a natural language query into a set of ontology entities and applies further algorithms for SPARQL generation. A machine learning technique is also applied to use feedback from users for improving the performance of SPARQL query construction. AquaLog [21] takes a query expressed in a natural language and translates it into query triples. These query triples will be matched with ontology/ semantically annotated information (i.e. RDF information). After the matching process, Aqualog will generate ontology triples which are the possible answers for users. Aqualog also includes the learning component for performance improvement similar to the FREyA's approach.

The challenge of the natural language interface is dealing with the various sources of ambiguities. Some query sentences are syntactically ambiguous while some are semantically ambiguous. A simple way to treat the ambiguities is by using dialogs. Querix [22] is a natural language interface system for querying ontologies based on clarification dialogs. Dialogs are used to solve the ambiguities in a user's query by asking a user to clarify the exact meaning of the words used. Similar to AquaLog, when a user enters a natural language query, the system decomposes the query into a query skeleton with the main word categories. The query skeleton is then matched with possible ontology's triples. When the system finds an ambiguity, it will show a menu with possible meanings to the user for selection.

The benefit of the natural language interface approach is similar to the semantic keyword search but the user input is in natural language. The challenge of this approach is disambiguation. The dialogs, as well as advanced natural language processing technologies, are needed to overcome this challenge.

V. THE COMMON IDEA OF SEMANTIC KEYWORD SEARCH ARCHITECTURE

Since this paper surveys semantic keyword search approaches. Firstly we will explain the common idea of semantic keyword search architecture.

Semantic keyword search approaches are based on the idea of automatically generating and selecting a set of formal queries derived from the keywords entered by a user. Basically, each approach provides the semantic keyword search mechanism and interface to access the semantically-enriched knowledge base for the discovery of information. The system will extract the possible meanings of the keywords from the domain specific knowledge base. It will then generate formal queries (i.e. SPARQL) and select the one (or a set of the formal queries) that is the best fits for the user requirements. Finally the formal query is executed and retrieves information from the knowledge base. The result of the search is returned to the user.

From the above concept, most approaches shared a common architecture as depicted in Fig.5. It consists of 2 main modules: pre-processing module and formal query construction module. The pre-processing module is to speed up the performance during the generation of formal queries. This module involves with indexing knowledge base entities. The formal query construction module is the focus of a semantic keyword search. Normally, it consists of 3 components: entity mapping, formal query generating and ranking.

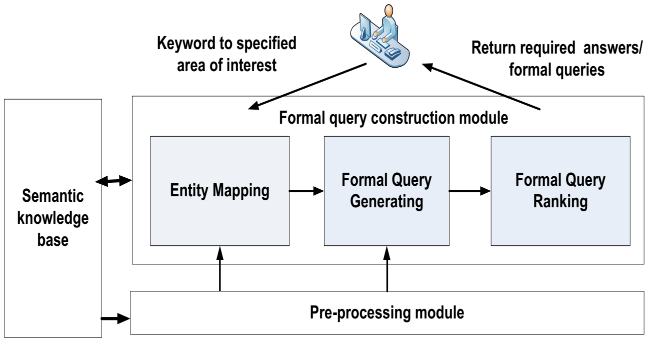
Keyword to specified area of interest
Return required answers/ formal queries
Formal query construction module
Semantic knowledge base
Entity Mapping
Formal Query Generating
Formal Query Ranking
Pre-processing module

Figure 5. The common architecure of semantic keyword search

The entity mapping component maps the keywords to the indexed knowledge base entities. The formal query generating component will then construct formal queries from the set of mapped entities. The techniques for constructing formal queries are different in each system. Some use a query graph technique while others use a template technique. This component will produce all the possible formal queries by interpreting the meanings captured by semantics (e.g. a class, data and object properties, a literal). The semantic query ranking component will rank and select the most relevant query (or the set of ranked queries) that matches the keywords entered by the user. In some system, the set of ranked query is directly forwarded to users. However, in some system the formal queries will be executed and return the results (answers) to the user.

VI. RELATED SEMANTIC KEYWORD SEARCH APPROACHES

This section reviews six semantic keyword search systems which shared the common architecture in section V. However, they were implemented by different approaches/techniques. These consist of : Semsearch [14], Quick [15], SPARK [7], Tran et al [3], Q2Sementic [4] and SKengine [16]. The approaches will be compared via criteria in next section.

Semsearch [14] proposes the use of predefined query templates to construct formal queries. The templates are a combination of all possible entity types from the knowledge base. The indexed semantic entities, including classes, properties and instances, are constructed to support the mapping of keywords to semantic entities. The input for indexing is the RDF data along with the ontology schema. Both the query templates and the indexed semantic entities are computed at pre-processing time. At runtime, each keyword term is mapped onto entities. All the mapped entities then are matched to the templates to construct the formal queries. After that all the formal queries are executed and retrieve results. The ranking engine will rank results and finally return to users.

Quick [15] also uses the predefined query templates to construct a set of possible semantic formal queries in a given domain. The process of constructing formal queries is the same as in Semsearch. It starts with a user entering keywords, then a guide is provided for supporting users to produce a semantic query step by step. This guide is in the form of a graph (including nodes and edges). The node represents the ontology's class that its instances are match with the keyword. The edge represents the object property that links two mapped classes. While this novel interface can help a user to construct a formal semantic query according to the user's choices, it is not suitable for users not familiar with graphs and the underlying concepts and relationships in the ontology.

SPARK [7] is a semantic keyword search approach that uses a graph based technique to construct formal queries. The SPARK framework starts at ontology processing which indexes entities in the knowledge base. The index is used for mapping user-input keywords with the indexed resources. The mapping step uses a string comparison technique and the semantic mapping using WordNet [23]. Those mapped entities corresponding to each keyword will be enumerated into query sets. The Kruskal's minimum spanning tree algorithm is then applied to construct a query graph for each query set. The algorithm explores the whole RDF graph and discovers the appropriate connecting nodes. Subsequently all possible query graphs are ranked by using the probability model and then will be transformed into SPARQL queries. Finally users will get the ranked list of SPARQL queries and select the one relevant to their information need.

Tran [3] proposes another approach for interpreting a keyword query using a semantic knowledge base. This approach uses a traversal graph algorithm to construct a query graph. After a user has entered the keywords, Lucene [24] is used to map those keywords with corresponding entities (e.g. class, property, literal, and instance). The query graphs are constructed by traversing the RDF knowledge base, and finding the neighbouring entities of each mapped entity within a limited range. After that, possible subgraphs which connect the mapped entities are extracted from the whole query graph. The different possible sub graphs are transformed to semantic formal queries and presented to users to choose the fit query one.

Q2semantic [4] is different from the above approaches. It supports a keyword search on schema-less RDF data graphs or those that do not include an initial ontology. Q2semantic uses an RDF graph clustering technique to infer an ontology structure (Clustered Rack Graph). An algorithm is used to generate top-k query graphs by exploring the ontology structure. The Q2semantic's query graph construction algorithm adopts the single-level search algorithm with distinct root nodes as discussed in Blink [25]. The Q2semantic ranking approach is used to generate the top-k of the query graphs. The top-k query graph will be transformed to a list of top-k SPARQL queries which allow people to select the most fit to their intention.

SKengine [16] is specifically designed for an expert discovery task. It also uses a graph based technique to construct formal queries. SKengine's pre-processing module consists of two types of indexes: (i) entity index is used to map corresponding entities to a keyword; (ii) ontology index will index relationships between the classes in the ontology. The ontology index is considered as a traversal space used in query graph construction. The query graph construction component will create query graphs from the set of mapped entities. The component will produce all the possible query graphs by interpreting the meanings captured by semantics. The fixed root node algorithm is undertaken to restrict the query graphs to a fixed root. The algorithm avoids the distinct root nodes. Since, it may generate irrelevant roots which are not related to the concept of expert. After that, all possible graphs are ranked by using 3 criteria: association length, entity mapping score and edge score. The graph with the highest ranked score will be transformed into SPARQL query. The SPARQL query will be executed and return the results to the user.

VII. Comparison of Semantic Keyword Search Approaches

This section presents a comparison of the semantic keyword search approaches reviewed in the previous section. The criteria for the comparison are proposed below. The criteria were derived from analysis of common ideas and technical implementations of the approaches.

A. The comparison criteria

1. Transparency: this criterion examines the user interaction with the system, and the following are the different types for investigation:

(i) Transparent: the semantic capability of the system is invisible to users. As a result, the system appears to users as an ordinary search engine. Users simply enter keywords into the system and the returned information is presented. Systems do not need to request additional information from users. RoundTrip ontology authoring [26] is an example of a transparent system. By using a text generator for constructing a controlled language for authoring ontology, it can hide the complexity of vocabulary/syntax of a controlled language as well as a formal semantic language from users.

(ii) Interactive: interactive systems ask users to clarify the meaning of keywords and take this into account for query construction.

(iii) Semi-transparent: this type of system also hides the semantic capability from users. It will automatically generate possible formal queries, however, ultimately the users have to select the queries which are relevant to their requirements.

2. Coupling: the above approaches are built on top of a semantic knowledge base. There are two levels of coupling between RDF information and ontologies in the knowledge base:

(i) Tight coupling: this refers to a semantic knowledge base with an explicit ontology schema and RDF information. The approach can directly uses ontology entities (e.g. class, property) and an ontology structure to support the creation of a formal query.

(ii) Loose coupling: this refers to a semantic knowledge base that obtains the schema-less RDF information. Linked data is an example of this type of RDF data. The approaches have to extract an implicit ontology structure from the RDF information at pre-processing time.

3. Pre-processing: these six approaches have a similar high-level architecture which consists of the pre-processing and query processing modules. In the pre-processing module, indexes are constructed to improve the performance of systems.

There are two types of index used to support the formal query construction:

(i) Entity index: this is an inverted file of ontology entities' labels and/or literals of any resource in the RDF data. The entity index is a common component to support the mapping of a keyword with mapped entities.

(ii) Ontology structure index: this type of index contains the structure of an ontology which is used to support the formal query construction based on a graph traversal approach. The index aims to improve traversal computation time. However, in a loose coupling system, other techniques may be used to extract the ontology structure, such as the Clustered Rack Graph in Q2Semantic.

4. Semantic formal query construction technique: Each approach provides different techniques for constructing a semantic formal query. With regard to the survey, these techniques can be classified into two main types: (i) using pre-defined query templates, and; (ii) a query graph construction.

The pre-defined query templates are only appropriate for a small number of keywords. When there are more keywords, the number of combinations will increase exponentially ending with a large number of complex patterns to be defined.

On the other hand, the query graph construction approach is more flexible. It does not need to fix the query structure to the form of the templates.

5. Graph traversal space: there are different algorithms for constructing a query graph. The algorithms can be categorised into two types based on the graph traversal space:

(i) Traversal of the whole RDF graph: the algorithms will traverse the whole RDF graph to construct query graphs. The main drawback of the algorithm is the computational cost because they require the exploration of the entire RDF data graph.

(ii) Traversal of a smaller space extracted by the use of ontology indexing or clustering of an RDF graph. It can reduce the traversing cost due to a much smaller traversal space.

6. Method of producing possible answers: Apart from traversing space classification, the query graph algorithms can be classified by the method for producing possible answers (i.e. the root of the query tree): (i) distinct root node; (ii) fixed root node.

In general, a formal query is based on an assumption of providing a form of query tree with a root. The root element is assumed to be the answer. For the distinct root node approach, the algorithm computes all possible query graphs with distinct roots. These provide a number of possible answers that might be relevant to users' intentions. The fixed root node algorithm aims to produce all possible query graphs with the same fixed root node. This kind of algorithm is for information searching in which users know exactly which kind of information they want and have fixed it for the answer of a semantic search.

7. Domain/task dependency: this refers to the coupling between algorithms in systems and a specific domain ontology/task. We can categorise systems into:

(i) Dependent: the algorithms in systems may be designed for specific a domain ontology/task such as an expert finding task.

(ii) Independent: the query construction algorithms are designed for general purposes, not specific to a domain ontology/task.

B. Comparison result

In Table I and Table II, we give an overview of the comparison results. Rows in both tables denote the seven criteria. Table I shows the comparison of the SemSearch, Quick and Tran at el approaches. The SPARK, Q2Semantic and SKengine are compared in Table 2. In case where it could gather ambiguous information for certain criteria, it will be denoted 'unclear' in the respective table entry.

The comparison results by transparency criterion show that SemSearch and SKengine are transparent systems. Since the systems appear to users as a traditional keyword search engine. Quick is an example of an interactive system because users take it into account for query construction. SPARK and Q2Semantic require users to select which relevant to their information need. As a result, they are semi-transparent systems. For the coupling criterion, we found most systems are built on top of knowledge base with explicit ontology schema. These do not include Q2Semantic because the system can deal with Linked data in RDF format.

SemSearch and Quick use a query template technique for query construction. On the other hand, the query graph construction techniques with different algorithms are applied to the other systems.

It was found pre-processing takes a significant role to improve the performance for all semantic keyword search approaches. As can be seen, all approaches need an entity index for mapping corresponded RDF entities to keywords. While techniques for capturing ontology structure (e.g. Clustered Rack Graph and Ontology index) are used in Q2Semantic and SKengine. The techniques aim to reduce traversal space during query graph constructions. However, techniques for traversal whole RDF graph are still used in Tran et al and SPARK.

With regard to domain dependency properties, the approaches (i.e. Tran el. Al, SPARK and Q2Semantic) were designed for general domain ontologies. They use the distinct root node algorithm to construct query graphs. This is for providing as many possible answers as to users. While an approach for a specific task such as SKengine requires a fixed root node algorithm. This is to reduce unnecessary answers and provides only a specifically fixed answer.

TABLE I. COMPARISON OF SEMANTIC SEARCH APPROACHES: SEMSEARCH, QUICK AND TRAN AT EL.

Criteria	SemSearch	Quick	Tran at el.
Transparency of background process to end user	Transparent	Interactive	Semi-transparent
Coupling of RDF information and ontology	Tight	Tight	Tight
Pre-processing	Entity index	Entity index	Entity index
Technique for query construction	Query templates	Query templates and query guide construction	Query graph construction
Graph traversal space	None	None	Whole RDF graph
Method of possible answers	Distinct root nodes.	Unclear	Distinct root nodes.
Domain dependency	Unclear	Independent	Independent

TABLE II. COMPARISON OF SEMANTIC SEARCH APPROACHES: SPARK, Q2SEMANTIC AND SKENGINE

Criteria	SPARK	Q2Semantic	SKengine
Transparency of background process to end user	Semi-transparent	Semi-transparent	Transparent
Coupling of RDF information and ontology	Tight	Loose	Tight
Pre-processing	Entity index	Entity index Clustered Rack Graph	-Entity index -Ontology index
Technique for query construction	Query graph construction	Query graph construction	Query graph construction
Graph traversal space	Whole RDF graph	Clustered Rack Graph	Ontology index
Method of possible answers	Distinct root nodes	Distinct root nodes	Fixed root node
Domain dependency	Independent	Independent	Dependent

VIII. SEMANTIC KEYWORD SEARCH EVALUATION METHODOLOGY

This section reports on the survey of evaluation methodology for semantic keyword search approaches. The evaluations concern on 3 mains features: effectiveness, efficiency and usability of the systems.

A. *Effectiveness*

A number of researches [2, 4, 8] investigated effectiveness of systems focusing on capability of formal query ranking component (see common semantic keyword search components in Section V). The ranking component creates a candidate list of constructed semantic queries in order by 'probability of acceptability'.

Fig. 6 illustrates the ranking effectiveness evaluation methods. To evaluate, set of keyword phrases (i.e. the input of systems) is provided. Keyword phrases may be extracted from problem scenarios or questions provided by experts/participants in the ontology domains. The manual semantic queries corresponded to the questions is created as a golden standard. The constructed semantic queries from systems were evaluated with the golden-standard manual queries. If the constructed semantic query is semantically equivalent to the golden standard query then it is 'acceptable'. The order of the acceptable query in ranked list is for evaluating the capability of ranking component.

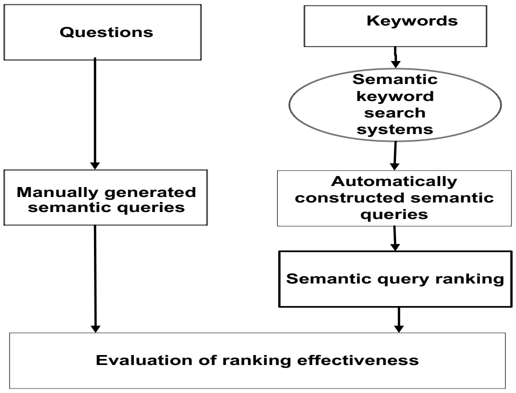
Questions
Keywords
Semantic keyword search systems
Manually generated semantic queries
Automatically constructed semantic queries
Semantic query ranking
Evaluation of ranking effectiveness

Figure 6. Methods of ranking effectiveness evaluation

The widely used metric to assess ranking capability is Mean Reciprocal Rank (MRR)[33]. The Reciprocal Rank (RR) is the multiplicative inverse of the rank of the first acceptable semantic query. Hence, the mean reciprocal rank is the average of the reciprocal ranks for all keyword phrases in the test set. MRR = $\frac{1}{|N_{KP}|}\sum_{i=1}^{N_{KP}}\frac{1}{rank_i}$ where N_{KP} = N_{KP} is the number of keyword phrases in the tested set, rank $_i$ = order of the acceptable semantic query of each keyword phrase. SPARK and SKengine used MRR metric for the evaluation.

Similarly, Q2semantic evaluation introduced a new metric named 'Target Query Position' (TQP). It also used to evaluate the effectiveness of semantic query ranking. Namely, TQP = 11 - P_{target}, where P_{target} means the position of the acceptable query in the ranked list. Note the higher the rank of the intended query, the higher its TQP score. If the rank of a query is greater than ten, its TQP is simply 0. Thus, the TQP score of a query range from 0 to 10.

B. *Efficiency*

The efficiency evaluation aims to gain a better understanding of the performance of systems. Parameters that would impact on system performance are investigated and the interested parameters of each system vary.

For example, Q2semantic investigated impact of size of search space (RDF graph) on the clustered graph index. In this regard, it was found out that the size of the clustered graph index depends heavily on the schema structure of the original RDF graph. Tran at al. [8] is semantic keyword search approach that provides top-k semantic queries with distinct root node. This efficiency evaluation studied the impact of parameter *k* on search performance. It could be observed that the search time increases linearly when *k* becomes larger. SKengine evaluation was concerned with the impact on its computational time by two parameters: (i) the number of keywords and (ii) the number of relationships between concepts in the ontology. The number of keywords is inversely related to the time performance of SKengine. Also, the higher number of relationships in the ontology causes the lower time performance.

In general, metric for efficiency evaluation is average performance time. The evaluator can use specific tools for time recording. For example, SKengine used the Netbeans profiler (http://profiler.netbeans.org), a module to provide a profiling functionality for the NetBeans IDE. The profiling function includes execution time profiling, which allows developers to be more productive in solving performance-related issues.

C. *Usability*

The usability evaluation investigates how the semantic keyword search systems are useful from users' point of view. Basically, the user interface of system is the focus of the study.

The presentation of semantic queries in easy way to users is necessary and it is a research challenge. However, from our investigation, there is likely related semantic keyword search work to study on the usability. Unfortunately, there was a study on usability of semantic search using natural language interface. Kaufmann and Bernstein [34] reported their usability study. They introduce four interfaces each allowing a different query language and present a usability study benchmarking these interfaces. The qualitative methods such as interview and questionnaires were used to capture the preference of users for each interface. It was found each interface have different weak/strong points in users' opinions.

IX. OUTSTANDING ISSUES

In this section, the summary of some open issues is reported. We are aware that these topics are by no exhaustive. It was concerned that our survey need further detail study. However, the following list reflects the outstanding issues expected to be important in future research.

A. *Disambiguation of keywords*

Ambiguities of user keywords are the main challenge for the semantic keyword search approaches. In particular, mapping keywords to associated knowledge base entities have to deal with various ambiguities. For examples: (i) a keyword term could be mapped with more than one entity due to its multiple meanings; (ii) users may use different keyword terms to identify the same concept in an ontology.

The uses of linguistic tools (i.e. Disco and WordNet) enable solutions that the semantic keyword search approaches should investigate. WordNet [23] has the potential to enrich the mapping with a set of synonyms. WordNet is a large lexical database for English language. It groups nouns, verbs, adjectives and adverbs into sets of synonyms which are provided, as well as general definitions and records of the various semantic relationships between these synonyms sets. In addition, a tool for computing semantic similarity between words (e.g. Disco) is also required. Disco [27] will be used for enriching the mapping process because it can provide similar words in the same context with any un-mapped keywords.

Even though, both tools have a potential to improve quality of mapping but they could not address all ambiguities. Therefore, the mapping techniques/tools that can reduce the ambiguities are still open issues.

B. *Understandable representation of semantics for users*

The semantic keyword/natural language search systems based on the interactive interface [15] attempt to address the challenge of the disambiguation by using dialogs or a query guide. The direct clarification of query meaning by users is appealing and it ensures that a system understands the user's intention exactly.

However, this poses a new challenge - how to represent and present the choices of meanings to users in an understandable manner. For example, Quick explains the meaning of candidate entities in the form of a graph which is difficult for users to understand.

Semi-transparent systems have a similar issue as the user has to select the most relevant query from a list presented. Unfortunately, the representation of the query candidates is in the form of graphs/pseudo-formal statements which make it difficult for ordinary users to determine the best query for their intentions.

C. *Cost reduction for formal semantic query construction*

The related work was designed for general tasks/domains so that the formal query construction module aims to produce all possible answers and query meanings that might be relevant for the user intention. However, for some specific tasks (such as expert finding), the type (concept) of the answer is known. Therefore, formal semantic query construction algorithms for generic purpose are unnecessary. Since, these usually search for a number of distinct types of answers and the user/system has to eventually spend extra cost to filter out irrelevant types of answers. The challenge is how to design a more efficient algorithm for the specific task.

D. *Adaptabiliy*

The semantic keyword search function is needed for many application domains. For example, (i) finding learning objects in e-learning domain; (ii) web services search in SOA architecture. However, it is an open problem how semantic keyword search can be adapted/adopted to those application domains. The plug-in component can support easy integration of semantic keyword search mechanism to the applications. This can reduce time to develop search function and provide user friendly interface to access information in RDF format.

E. *Ranking*

A ranking scheme is required to return the queries that most likely match the user intended meaning. Ranking has been dealt with the semantic keyword search approaches. There are many criteria for ranking such as length of the formal query, mapping score, the importance of node in query graph and so on. In regard to adaptation of semantic keyword search to other application domains, weight adjustment of each ranking criteria is a challenge. The different applications may be suitable for different criteria and varied weight.

X. Conclusion

In this work, we introduced a comparison of six semantic keyword search approaches. With regard to our comparison criteria, the common ideas and implementation techniques were explained. In addition, the advantages and limitations of each technique were briefly discussed. The evaluation methodology was presented to guide researchers on how to conduct experiment evaluation. Finally, we identified research and application-development issues that need to be addressed by the current systems.

From this study, a number of promising approaches were explored. However, for the areas to mature it takes two crucial requirements. Firstly, the research community has to fill gaps, discussed in section IX. Secondly, it requires researchers to deeply investigate factors that could impact to the effectiveness and performance of the systems. The concern of the factors will be useful for improving efficient design of the architectures and algorithms in the semantic keyword search approaches further.

Reference

[1] Antoniou, G. and F.v. Harmelen, *A semantic web primer*. 2004, Massachusetts,USA: The MIT Press.

[2] Damljanovic, D., M. Agatonovic, and H. Cunningham, *Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-Based Lookup through the User Interaction*, in *The Semantic Web: Research and Applications 7th Extended Semantic Web Conference, ESWC 2010*. 2010, Springer.

[3] Tran, T., et al., *Ontology-based Interpretation of Keywords for Semantic Search*, in *ISWC/ASWC.*2007: Busan, Korea. p. 523-536.

[4] Wang, H., et al., *Q2Semantic: A Lightweight Keyword Interface to Semantic Search.* 2008.

[5] Corby, O., R. Dieng-Kuntz, and C. Faron-Zucker. *Querying the Semantic Web with Corese Search Engine*. in *The 15th ECAI/PAIS*. 2004. Valencia, Spain.

[6] Cheng, G. and Y. Qu, *Searching Linked Objects with Falcons: Approach, Implementation and Evaluation.* International Journal on Semantic Web and Information Systems, 2009. **5**(7): p. 50-71.

[7] Zhou, Q., et al., *SPARK:Adapting Keyword Query to Semantic Search*, in *ISWC/ASWC*. 2007: Busan, Korea. p. 694-707.

[8] Tran, T., et al. *Top-k Exploration of Query Graph Candidates for Efficient Keyword Search on RDF*. in *Proceedings of the 25th International Conference on Data Engineering (ICDE'09).*2009.

[9] Heflin, J. and J. Hendler. *Searching the Web with SHOE*. in *The AAAI workshop on AI for web search*. 2000: AAAI Press.

[10] Bizer, C., T. Heath, and T. Berners-Lee, *Linked data - the story so far.* Journal on Semantic Web and Information Systems, 2009.

[11] Berners-Lee, T., et al. *Tabulator: Exploring and Analyzing linked data on the Semantic Web*. in *The 3rd International Semantic Web User Interaction Workshop(SWUI06)*. 2006.

[12] Kobilarov, G., et al. *Media Meets Semantic Web – How the BBC Uses DBpedia and Linked Data to Make Connections*. in *The 6th European Semantic Web Conference(ESWC2009)*. 2009.

[13] Prud'hommeaux, E. and Andy Seaborne. *SPARQL Query Language for RDF.* 2008 [cited 2009 February]; Available from: http://www.w3.org/TR/rdf-sparql-query/.

[14] Lei, Y., V. Uren, and E. Motta. *Semsearch : A seach engine for semantic web*. in *Proceeding of the 15th International conference on Knowledge Engineering and Knowledge management(EKAW)*. 2006.

[15] Zenz, G., et al., *From keywords to semantic queries incremental query construction on semantic web.* Web Semantics:Science, Services and Agents on the World Wide Web, 2009. **7**: p. 166-176.

[16] Sitthisarn, S., L. Lau, and P. Dew. *Semantic keyword search for expert witness discovery*. in *STAIR'11: International Conference on Semantic Technology and Information Retrieval*. 2011. Kuala Lumpur, Malaysia.

[17] Fazzinga, B. and T. Lukasiewicz, *Semantic search on the web.* Semantic web, 2010. **1**: p. 89-96.

[18] Strasunskas, D. and S.L. Tomassen, *A Role of Ontology in Enhanceing semantic search: the EvOQS Framework and its Initial Validation.* International journal of Knowledge and Learning, 2009. **4**(4): p. 398-414.

[19] Castells, P., M. Fernandez, and D. Vallet, *An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval* IEEE Transactions on Knowledge and Data Engineering 2007. **19**(2): p. 261-272.

[20] Bhagdev, R., et al. *Hybrid search: effectively combining keywords and semantic searches*. in *The 5th European semantic web conference on The semantic web: research and applications* 2008: Springer-Verlag Berlin.

[21] Lopez, V., Pasin, M. and Motta, E. *AquaLog: An Ontology-Portable Question Answering System for the Semantic Web*. in *Proceedings of European Semantic Web Conference (ESWC 2005)*. 2005.

[22] Kaufmann, E., Bernstein, A and Zumstein, R. *Querix: A Natural Language Interfaceto Query Ontologies Based on Clarification Dialogs*. in *5th International Semantic Web Conference (ISWC 2006)*. 2006: Springer.

[23] WordNet. *WordNet: A lexical database for English*. 2011 [cited 2012]; Available from: http://wordnet.princeton.edu/.

[24] *Lucene*. 2011 [cited 2011 17 November 2011]; Available from: http://lucene.apache.org/java/docs/index.html#Apache%20Lucene.

[25] He, H., et al., *BLINKS: Ranked Keyword Searches on Graphs*, in *SIGMOD Conference*. 2007.

[26] Brian Davis, A.A.I., Adam Funk, Valentin Tablan, Kalina Bontcheva, Hamish Cunningham and Siegfried Handschuh, *RoundTrip Ontology Authoring*, in *The Semantic Web - ISWC 2008*. 2008. p. 50-65.

[27] Disco. A linguatools. 2011 [Cited 2012]; Available from: http://linguatools.de/disco/disco_en.html.

[28] Manola, F. and Miller, E. 2004. *RDF Primer* [Online]. Available: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/ [Accessed 5th May 2008 2008].

[29] Gruber,T.R.1993. A translation Approach to Portable Ontology Specifications. *Knowledge Acquistion,* 5, 199-220.

[30] Genesereth, M.R. and Nilsson, N.J. 1997. *Logical Foundations of Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann.

[31] Wu, J. and Yang, G. 2005. An Ontology-Based Method for project and Domain Expert Matching. In *FSKD 2005, LNAI 3614.* Springer-Verlag Berlin Heidelberg.

[32] Liu, P. and Dew, P. Using Semantic Web Technologies to Improve Expertise Matching within Academia. I-KNOW '04, 2004, Graz, Austria.

[33] Voorhees, E. M. 2000. The TREC-8 Question Answering Track Report. NIST.

[34] Kuafmann, E. and Bernstein, A. Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. Web Semantic: Science, Services and Agents on the World Wide Web, 2010. 8: p. 377-393.