Abstract—Semantic web search engines are evolving and manyprototype systems and some implementation have been developed. However, there are some different views on what a semanticssearch engine should do. In this paper, a categorization scheme for semantic web search engines are introduced and elaborated. For each category, its components are described according to a proposed general architecture and variousapproaches employed in these components are discussed. We also propose some factorsto evaluate systems in each category.

I. INTRODUCTION

This paper tries to analysis semantic search engines andprovides a rational categorization scheme for them. To the bestof our knowledge, in this regards there is no work reported. However, there is only a short explanation in [19]. According to [5], semantic web (hereafter is referred to as SW) has some distinguishing features that affects search process:

- Instead of web documents, in the SW, all objects of thereal world are involved in the search.
- Information in SW is understandable by machines as well as human.
- SW languages are more advanced than html. It is possible to distribute information about a single concept in SW. Therefore semantic search engines have following fundamental differences to the traditional search engines:

Using a logical framework lets more intelligent retrieval possible.
- There are more complex relations in documents resulting in the importance of the problem of meta-data maintenance, update and more complex ranking.
- Specifying relationships among objects explicitly highlightsthe need for better visualization techniques for theresults of a search. One important aspect of SW search is the usage of ontology and meta-data. Ontology provides explicit conceptualization for entities in a specific domain. Another important aspect is the annotation for the current web pages. Annotations are meta-data useful for machines to understand the content of a web page. In such meta-data concepts are pointers to already-defined ontologies [24]. Respecting the kind of search in SW, it is possible to categorize users to two groups.

One group are ordinary users that do searching as like the current web but demand more accurate and complete results than traditional search engines. Second group are application developers in the SW where their primary goal is to search and retrieve SW documents. According to these two categories of users, we can categorize SW search engines to the following two categories:
- Engines specific to the SW documents: they search only documents that are represented in one of the language specific to SW.
• Engines that tries to improve search results using SW standards and languages. Using context information (represented by domain ontology and metadata) is one of the key aspects for these engines.

The paper is organized as follows. In section 2 different types of annotation and methods of generating them are discussed. Section 3 provides explanations about search engines for developers and advanced users in SW. Search engines that use SW concepts to provide better search results are discussed and further categorized in section 4. For each category, in section 5, we have a brief analysis and propose some points for evaluating them. Section 6 concludes the paper by providing a summary of all reported works in SW search.

II. ANNOTATION METHODS

One of the major problems facing researchers in SW is annotation which is a prerequisite for SW search engines. To adopt current web pages for SW search engines they should be annotated by finding appropriate meta-data to be added to each one. There are different approaches which spawn in a broad spectrum from complete manual to full automatic methods. Selection of an appropriate method depends on the domain of interest [24]. In general meta-data generation for structured data is simpler [24]. Annotations can be categorized based on following aspects:

• Type of meta-data: According to [25] meta-data can be divided to two types of Structural and Semantic. In the former, non contextual information about content is expressed (e.g. language and format). In the later, the main concern is on the detailed content of information and usually is stored as RDF triples.

• Generation approach: a simple approach is to generate meta-data without considering the overall theme of the page and only using structural information of a page together with natural language processing techniques. A better approach is to use an ontology in the generation process. In this case it is possible to use clustering methods to distinguish the general type of a page [25]. Then using a previously specified ontology for that type, generate meta-data that instantiates concepts and relations of ontology for that page. The main advantage of this method is the usage of contextual information.

• Source of generation. The ordinary source of meta-data generation is a page itself but sometimes it is beneficial to use other complementary sources. For example [1] and [5] discuss about using network available resources for accumulating more information for a page. For example for a movie it might be possible to use IMDB to extract additional information like director, genre, etc.

Although there is no complete reference about meta-data but [18] provides a rather complete list of systems that generate meta-data.

II. ONTOLOGY SEARCH ENGINES

According to [1] and [8], for the following reasons, it is not possible to use current search engines for SW documents:

• Current techniques does not let to index and retrieve semantic tags.
• They don’t use the meaning of tags
• Can’t display results in visual form
• Ontologies are not separated entities which usually have cross references that current engines don’t process. In general there are two approaches to handle these documents: using current search engines with some modifications or creating a special search engine. In what follows each of these approaches is further elaborated.

A. Ontology Meta Search Engines
This group do retrieval by putting a system on top of a current search engine. There are two types of this systems.
In the first type, there is a search engine that only searches specific file types (e.g. RSS, RDF, OWL). The main concerns of such systems are on the visualization and browsing of results. For example in [8] an engine forwards a user’s request for a specific file type to Google search engine and then using a visualization tool lets user to navigate and display results.
In the second type there is possible to search on semantic tags. But since those tags are ignored by the underlying search engine, an intermediate format for documents and user queries are used. In [2] a technique named Swangle is issued for this purpose. With this technique RDF triples are translated into strings suitable for underlying search engine. For example consider the following triple which is in n3 Notation.

```
Each of these terms are converted to a string and added to the document for indexing. On the other side, this translation is done for user queries too.
```

B. Crawler Based Ontology Search Engines
These engines use a specific crawler for SW documents.
In the figure 1 one complete system is shown [4]: Architecture of a SW document specific search engine. Here, based on the four sections specified in the architecture, Analysis of them is given.

1) Discovery: Crawling of SW documents are different from HTML documents. Actually they are knowledge crawlers which are more complex than traditional ones [27]. In SW we express knowledge using URI in RDF triples. Unlike HTML hyperlinks, URIs in RDF may point to a non-existing entity. Also RDF may be embedded in HTML documents or be stored in a separate file. Such crawlers should have the following properties [27]:
• Should crawl on heterogeneous web resources (owl, oll, daml, rdf, xml, html).
• Avoid circular links
• Completing RDF holes
• Finding new semantic web documents from information in the currently under
process document (e.g. Extend and Import specifications).

<table>
<thead>
<tr>
<th>Priority</th>
<th>Relationship</th>
<th>Language Specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>instantiation</td>
<td>rdf:type</td>
</tr>
<tr>
<td>2</td>
<td>subClass</td>
<td>rdf:type, subClass, domain: subClass</td>
</tr>
<tr>
<td>3</td>
<td>domain:range</td>
<td>rdf:type, domain, domain: range</td>
</tr>
</tbody>
</table>

**TABLE I**

RELATIONSHIP TYPES

In [26] crawling of SW documents is explained in detail. According to [27] it is possible to categorize the derived ontologies based on a clustering method. For indexing and retrieval of SW documents N-grams and bag of URI refs are proposed. More explanation and comparison of them are given in [3]. Three types of meta data are used in these systems [3]:

- Language attributes
- Relationships between SW documents (prior version, imports, extends, etc.)
- Meta data resulting from analysis

2) Analysis: For offline ranking it is possible to use the references idea of PageRank. But the main point is that count, type and meaning of relations in SW is more complete than the current web. In the table I three types of relation and their corresponding values are specified [27]: In [4] Onto Rank values for each ontology is calculated very similar to PageRank in traditional search engines like Google.

3) Digest: In a recent version this algorithm is similar to what is given in [27] which uses a sum of rank and priority of concepts in a SW document to calculate the overall rank of a document. 4) Service: In addition to user interface in this section, services to application systems are provided too.

III. SEMANTIC SEARCH ENGINE

Searching in the web is done either using search engines or web directories, each having respected restrictions [5]. One interesting example is shown in [5]: if we search for “Matrix”, non homogeneous results ranging from mathematical matrix, Matrix movie and so on is returned. Semantic web is introduced to overcome such problems. The most important tool in semantic web for improving search results is context concept and its correspondence with ontologies. This type of search engines uses such ontological definitions. It is possible to categorize this type of search engines to three groups. In the first group which is the largest one, aim is to add semantic operations for better results.

In the second group, using facilities of semantic web the goal is to accumulate information on a topic we are researching on. Search engines in the third group try to find semantic relations between two or more terms.

A. Context based Search Engines

Figure 2 shows an overall framework for this kind of engines. It should be emphasized that very limited number of engines have all of the functionalities specified in the figure.

1) Crawling the semantic web: There is not much difference between these crawlers and ordinary web crawlers and in fact many of the implemented systems use an existing webcrawler as underlying system. For example in [1] haircut is used as underlying system and also [15] uses
one that understandsspecial semantic tags. One of the important features of these crawlers should be the exploration of ontologies that are referred from existing web pages.

2) Metadata generation: According to discussion in section 3, there are different ways for metadata generation. For example, [1] and [5] use external metadata. [1] Uses Aero Textto extract names and expressions and then generates metadata in RDF format. One of the important problems in this regards is semantic normalization [25] meaning to generate metadata for different resources in same form. For example [12] is an anon-standard example in which metadata is represented in adhoc model. In semantic portal [23] producers should generate annotations and there is a summarization and collection of metadata in the central server. As explained before, metadata generation is simpler and more accurate when the theme of a page is known. For example in [15] using a tool named Knowledge Annotator terms of ontology is used to describe information in a given page. Also in [18] proposed a method for generating and managing metadata according to already defined ontologies. But if ontology for a page is not known in advance, its possible to use clustering techniques like what explained in [25] to find an appropriate ontology. Knowledge Parser [24] is a kind of complete system using important techniques from different areas like NLP, Text Engineering, Document Structure Processing, and Layout Processing. Its operation is shown in the figure.

3.3) Indexing: Most of the engines do not provide any special functionality regarding indexing. OWLIR [1] uses Swangling explained earlier. [6] Introduces Ontological Indexing in which indexing is done based on a reference ontology. Also in [18] possibility of dividing documents to smaller parts is used to improve indexing performance. Also in p2p architecture of [22] for each of concepts in the reference ontology there exist an agent that maintains information corresponding to it.

4) Accepting user’s requests: There are two different approaches: term-based and form-based. In term-based approach used in [5], [23], and [24], it is tried to find the search context from entered keywords. In the form-based approach used in [1], [15], [23], and [24], user interface is generated according to the ontology selected by user.

5) Generating metadata for user requests: This operation is very similar to generating metadata for documents. For example in [18] the same Semantic Mapper is used for generating metadata both for documents and user requests. Often Wordnet is used to expand user requests. For example in [20] for termed entered by a user, using Wordnet, synonyms are found and used to expand the query.

6) Retrieval and ranking model: Usually an ordinary VSM model [30] is used and then based on RDF graph matching.
results are pruned. In [9] from the equivalence of RDF graphs and Conceptual Graphs (CG), already existing operations on CGs is used to match user request and documents.

Semantic Distance concept is often used to estimate similarity of concepts in a matching process. In [21] this measure is defined for different elements in graphical representations. It is also possible to use graph similarity for ranking results. However, in [7] a fuzzy approach is used for this purpose.

7) Display of results: A major difference of semantic search engines and ordinary ones is the display of results. One of the primary tasks is to filter the results (for example for eliminating repetitions). In [6] in addition to normal display of results, a number of classes is displayed and when a user selects one, only those results having instances of the classes is shown. In [23] display is a kind of hierarchy in which top concepts of ontology are shown and by selecting one, detail of it according to the ontology is displayed.

B. Evolutionary Search Engines

The advanced type of search is something like research; in fact as mentioned in [5] this kind of searches aim at gathering some information about a specific topic. For example if we give the name of a singer to the search engine it should be able to find some related data to this singer like biography, posters, albums and so on. These engines usually use one of the commercial search engines as their base component for searching and then augment returned result by these base engines. This augmented information is gathered from some data-insensitive web resources. In figure 4 we showed overall architecture for such engines. As it can be deduced from the figure this architecture has some similarities with what we discussed in previous subsection. Here we crawl and generate annotation just for some well known informational web pages i.e. CDNNow, Amazon, IMDb as mentioned in [2] and [5]. After this phase we collect annotations in a repository. Whenever a sample user posed a query two processes must be performed: first, we should give this query to a usual search engine (usually Google) to obtaining raw results. Second, the system will attempt to detect the context and its corresponding ontology for the user’s request in order to extract some key concepts. Later we use these concepts to fetch some information from our metadata repository. The last step in this architecture is combining and displaying results. Main problems and challenge in these types of engines are [5]:

Fig. 2. Semantic Search Engines’ Architecture

Fig. 3. Annotation Generation Steps in KnowledgeParser
C. Semantic Association Discovery Search Engines

Usually one of the user’s interests is finding semantic relations between two input terms. Old search engines handled these requests using learning and statistical methods [25], but semantic web standards and languages have provided more effective and precise methods. SemDis [10],[14] is a real sample for these systems, its goals is finding and ranking semantic associations. Overall architecture of SemDis is shown in figure 5. There are different types of semantic association but most known of them is a sequences of classes and relations between two classes. In fact we talk about just two terms because as said in [13] average length for users’ queries is 2.3 term. With respect to our definition for semantic association, two terms may have one of these association: Null (both of them are instances of one concept), Direct (when there is a direct relation between them) and Indirect (chain of relations instead of single direct one). In the [13] Bayesian network was applied in order to discover semantic association. Our reference ontology forms the graph of this network and logs of user’s queries are used to computing its parameters. In general manner, for finding semantic association between more than two terms some techniques have been proposed, for example in [16] Spread Activation Technique is used to expand an initial set of instances to contain most relative instances to them. The initial set is populated by extracting important terms from user’s query, then with respect to the metadata repository corresponding instances is retrieved and after expanding them an instances graph is produced which each of its edges has correctness weight in addition to usual semantic label. Technically speaking, after discovery phase often we have numerous semantic associations, therefore a ranking policy must be used. In [10] some criteria for these ranking algorithms are introduced:

- **Context:** special part of reference ontology that is interested by user
- **Subsumption:** low level classes in hierarchy have more information than their parents
- **Path Length:** having a shorter path between two terms indicates that they have near meaning
A. Ontology Search Engines

In contrast to usefulness of meta-search engines for regular pages in traditional web, it seems that they are not so good for ontologies. In fact we can not collect all ontologies in the web just but using filetype command within commercial search engines. In addition swangling operation has a huge amount of overhead, therefore it’s much better to use crawler-based ontology search engines (2nd category) rather than ontology meta-search engines (1st category). In order to evaluating performance of this kind of search engines there is no standard test collection, but we can simply test them by searching for ontologies using name of ontologies, classes and properties and judge their results according to the precision measure (portion of relevant result from all result returned)

- Trust: obtained results from trusted resources is more valuable in final ranking results

B. Semantic Search Engines

1) Context-Based Semantic Search Engines: The main strangeness of these engines is their simplicity. In fact because they tried to be as simple as textbox search engines (like google) they are most popular search engines in the semantic web. Here quality of results heavily depends on power of its annotation module. The biggest problem of these search engines is that they are limited to the special contexts. It is much better if we can develop a multi-contextual semantic search engine. Fortunately we can apply standard measures (i.e. Precision and Recall) and standard test collections (i.e. TREC tracks) of traditional information retrieval to evaluating this kind of semantic web search engines. It should be noted that if we can prepare ontology for test documents, the results will show much improvements.

VI. CONCLUSION

Compared to other categories, the semantic association discovery engines are related to higher layers of semantic web cake (logic and proof). Result of these engines is very depending on their ontology repository. For evaluating them we can use an upper ontology like WordNet, after selecting two concepts randomly, the correctness and speed of discovering paths between them are two useful measures for performance evaluation.

REFERENCES


