Search Engine For Semantic Web

1. Introduction: The good news about the Internet and its most visible component, the World Wide Web, is that there are hundreds of millions of pages available, waiting to present information on an amazing variety of topics. The bad news about the Internet is that there are hundreds of millions of pages available, most of them titled according to the whim of their author, almost all of them sitting on servers with cryptic names. When you need to know about a particular subject, how do you know which pages to read? If you're like most people, you visit **Search Engine**.

Internet search engines are special sites on the Web that are designed to help people find information stored on other sites. There are differences in the ways various search engines work, but they all perform three basic tasks:

They search the Internet -- or select pieces of the Internet -- based on important words. They keep an index of the words they find, and where they find them. They allow users to look for words or combinations of words found in that index. Early search engines held an index of a few hundred thousand pages and documents, and received maybe one or two thousand inquiries each day. Today, a top search engine will index hundreds of millions of pages, and respond to tens of millions of queries per day.

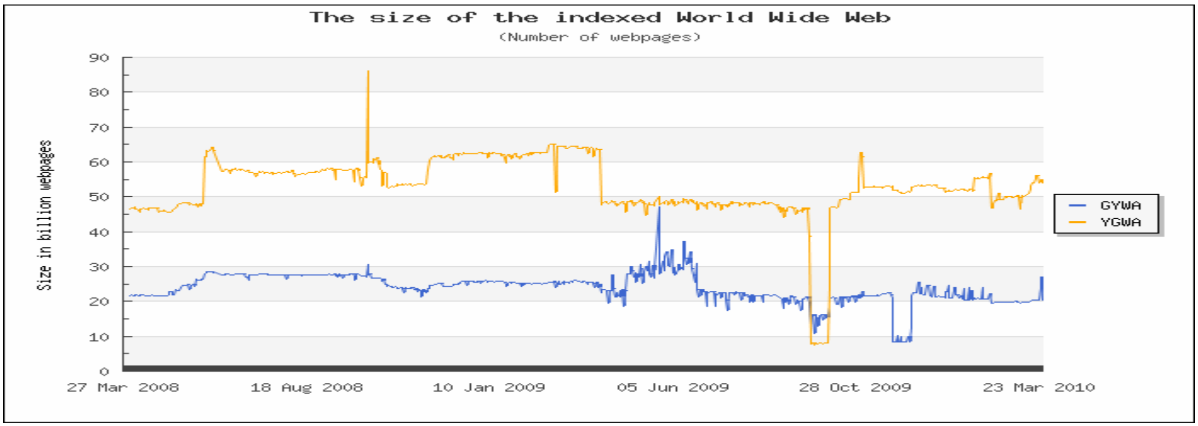
2. Current Web: The Internet’s roots begin with ARPANET, a project commissioned by the Advanced Research Projects Agency (ARPA) to study countrywide data communication. In 1990, Tim Berners-Lee developed the first version of his World Wide Web program at CERN. He wrote the first Web browser and Web server and the protocols that they used to communicate. During theses years billions pages were linked into his Web, more users grabbed software to use it. Enormous amount of information is available where user can navigate to other documents via links. Documents are accessed via unique address of each document called Uniform Resource Locator. These document are generally HTML documents. Current web is highly decentralized, dynamic, vast and contains information that is widely used only for displaying.

Though HTML is mainly for rendering information, it provides two tags for describing information that it displays. First is META. The META element specifies meta-data in the form of a name/value pair. A popular use for META was to indicate keywords, for example <META name=”keywords” content=”Semantic Web”>, that could help search engines index the site. However, many sites began abusing the keywords by including popular keywords that did not accurately describe the site (this is known as keyword spamming). As a result, many search engines now ignore this tag. The REL attribute of the anchor (<A>) and link (<LINK>) elements names a relationship from the enclosing document to the document pointed to by a hyperlink.

Despite its popularity, HTML suffered from two problems. First, whenever someone felt that HTML was insufficient for their needs, they would simply add additional tags to their documents, resulting in a number of non-standard variants. Second, because HTML was mostly designed for presentation to humans, it was difficult for machines to extract content and perform automated processing on the documents. To solve these problems, the World Wide Web Consortium (W3C) developed the Extensible Markup Language (XML)

3. **Need For Searching:** Due to rapidly growing technologies, size of web is increasing tremendously. To find exact match of required information without searching techniques is not possible.

4. **Size of Web:**

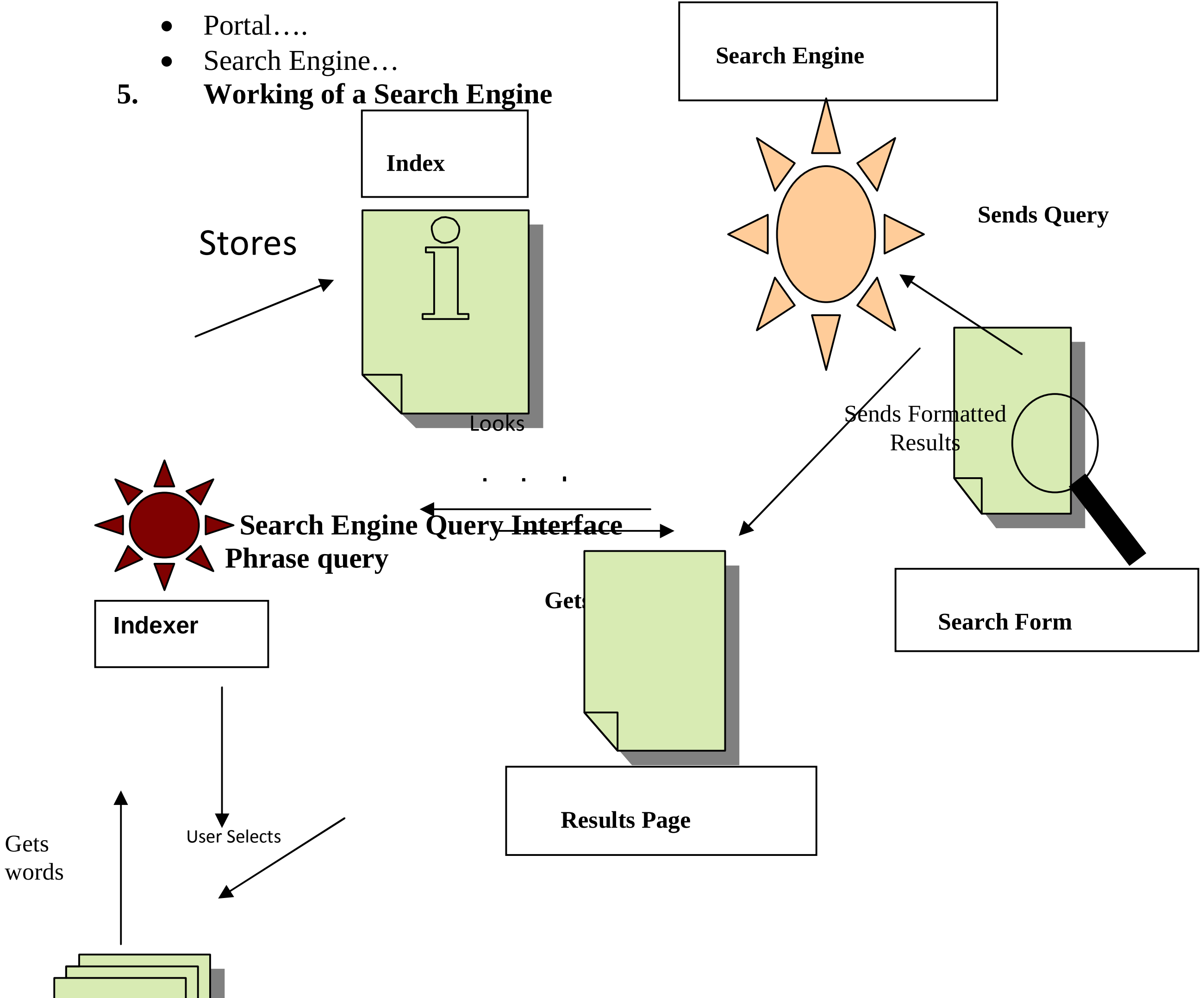


GYBA = Sorted on Google, Yahoo!, Bing and Ask
YGBA = Sorted on Yahoo!, Google, Bing and Ask

Thus to cope with large volumes of Data various Searching Methods have been developed, namely:

- Directories …
- Portal….
- Search Engine…

5. **Working of a Search Engine**



A phrase query represents a query that is matched against a consecutive sequence of terms in the field. For example, the phrase query 'winding road' should match 'winding road' but not 'road winding'.

Boolean query

A Boolean query represents a composite query that may contains sub queries of arbitrary nesting level and with composition rules such as 'and', 'or' or 'not'.

Wild cards

A wild card is the special characters that can appear at the end of a word so that you can search for all possible endings to that root. For example, if you are looking for information on the national things. Documents, which contain the following words, may all be useful to your search: nation, national, nationality, etc. If your search engine allowed wild cards, you would enter "nation*". In this case, the asterisk is the wild card and documents, which contained words that started with “nation”, would be returned.

6. Narrowing the search: The Boolean operator "and" is the most common way to narrow a search to a manageable number of hits. For example, with "heart and disease" as the search term, an engine will provide links to sites which have both of these words present in a document. It will ignore documents which have just the word "heart" in it (e.g., heart transplant) and it will ignore documents which have just the word "disease" in it (e.g., lung disease, disease prevention). It will only make a link if both of the words are present - although these do not necessarily have to be located beside each other in the document. For even more narrow searches, you can use "and" more than once. For example, "heart and disease and prevention" would limit your search even more since all three terms would have to be present before a link would be made to the document.

The Boolean operator "not" narrows the search by telling the engine to exclude certain words. For example, the search term "insecticides not DDT" would give you links to information on insecticides but not if the term "DDT" was present. It is possible to combine two different operators. For example, the term "endangered and species not owl" would give you information on various kinds of endangered species - both of the words "endangered" and "species" would have to be present for there to be a hit. However, you would not get information on any owls that are endangered since the "not" term specifically excludes that word.

7. Widening the search: The Boolean operator "or" will broaden your search. You might use "or" if there were several words that could be used interchangeably. For example, if you were looking for information on drama resources, using just that one search term might not give you all that you wanted. However, by entering "drama or theater", the search engine would provide a link to any site that had either of those words present. For even wider searches, you can use "or" more than once. For example, "drama or theater or acting or stage" would provide a very broad search indeed.

8. Boolean operators: The short answer to that question is - "not consistently". Some engines allow the use of just a few operators while others provide access to a wide range. Some require you to enter the operator yourself while others have you select the operator from a pop-up box. Some allow you to do any kind of search from the main search page while others require you to go to an "advanced" page to conduct boolean searches. Some engines allow you to enter several words into the search term WITHOUT a boolean operator. However, some search engine will assume that there is an "or" operator between the words while others assume the desired operator is "and". Check what the engine's default operator is before you elect not to enter boolean operators. Most search engines are pretty easy to use if you read their help information.

9. "Meta" search engine : A Meta search engine is a search tool that doesn't create its own database of information, but instead searches those of other engines. "Metacrawler", for instance, searches the databases of each of the following engines: Lycos, WebCrawler, Excite, AltaVista, and Yahoo. Using multiple databases will mean that the search results are more comprehensive, but slower to obtain. **Making sense of the user query.** As mentioned earlier in , making sense of the user query is the first step of the search process in SemSearch, whose task is to find out the semantic meanings of the keywords specified in a user query so that the search engine knows what the user is looking for and how to satisfy the user query.From the semantic point of view, one keyword may match i) general concepts (e.g., the keyword “phd students” which matches the concept phd-student), ii) semantic relations between concepts, (e.g. the keyword “author” matches the relation has-author), or iii) instances entities (e.g., the keyword “Enrico” which matches the instance Enrico-Motta, the keyword “chief scientist” which matches the values of the instance Marc-Eisenstadt of the property has-job-title). The ideal goal of this task is to find out the exact semantic meaning of each keyword.

This is however not easy to achieve, as there may be more than one semantic entity which matches a keyword. Thus, we relaxed the goal as finding out all the semantic entity matches for each keyword. For the purpose of finding out semantic entity matches, we used the labels of semantic entities as the main search source. The rational for this choice is that from the user point of view labels often catch the meaning of semantic entities in an understandable way. In the case of instances, we also used their short literal values as the search source. So that when the user is searching for “chief scientist”, the instance that has such a string as a value of its properties can be reached.

In order to produce fast response, the search engine first indexes all the semantic entities contained in the back-end semantic data repositories, including classes, properties, and instances. It then searches the indexed repository to find out matches for keywords. Thus, two components are developed in the search engine, namely the semantic entity index engine and the semantic entity search engine. As it narrows the search sources to labels and short literals of semantic entities, the search engine is able to find out semantic entity matches for each keyword. These matches are the possible semantic meanings of keywords. Please note that for the sake of getting quick response, we only use text search to find string matches for user keywords at the moment. We avoid using techniques like WordNet [3] based comparison to find matches. This might cost us some good matches, e.g., losing the match table if the user is searching for desk. But one to one comparison is time consuming and expensive in real-time scenarios. This is indeed a trad-off as well as a research challenge that we need to address in future.

10. Translating the user query into formal queries: In this step, the search engine takes as input the semantic matches of user search terms and outputs an appropriate formal query according to the semantic meanings of keywords. To achieve this task, the search engine needs to capture the focus of the user query (i.e., the type of the expected search results). As described earlier in Section 4, the subject keyword specifies the type of the expected search result. Thus, it is reasonable to expect that the queried subject is a general topic or concept (i.e. class). In the case when the subject keyword does not match any class, the search engine needs to figure out what the expected results are. This will be discussed in the following subsections. To better understand how to construct formal queries from user queries, we classify user queries into two types: i) simple queries which only comprise two keywords, and ii) complex queries where more than two keywords are involved. In the case of simple queries where the types of semantic entity match combinations are fixed, we developed a set of templates to support the formulation of formal queries. The situation is much trickier in the case of complex queries where there are many variables for keywords combinations.

In this section, we first look at the formulation of formal queries from simple user queries. We then investigate how to handle complex ones. As we used the Sesame SeRQL language3 as the formal query language in the prototype of the SemSearch search engine, we explain the mechanism using the same language

(Please note the underlying approach does not confine itself to any specific query language).

11. Some Factors Which May Affect Search Engine Ranking: Domain Extension - the search engines do not always immediately recognize new extensions. This was a problem for .cc and .biz sites in the early going.

Sub domains - If your web site is 'mysite.network.com, and 'network.com' has engaged in any unsavory search engine spamming, your site will be affected.

IP Address/Range - This is a bit like the last point. If the search engines have had problems with many sites from one hosting company, they may degrade all the sites from that company’s IP range. It makes the hosting companies behave.

"Domain in use since" The longer it's live the better it's generally viewed. Kind of a respect thy elders thing...

12. Negatives That Affect Your Position Within The Search Engines: Broken links - Internally and outgoing.

Spam

Metatag Stuffing.

Irrelevance - If you use irrelevant keywords, description, etc.

Tiny Text. - If you use text that is too small for the eye to see.

Invisible Text - Text the same color as the background.

Meta Refresh Tag

Redirects - Where when you try and get to one page, but the address changes to a different one.

Excessive Search Engine Submission - over submitting may get your site banned.

Frames - Be careful when you use them. You need to embed key terms in them, because generally, the search engines can only see the frame, and not the primary content that you see as visible.

Empty Alt Tags - Leaving these empty shows is poorly viewed. It's akin to bad coding.

Compounded Words in the content or tags will not help the web site for individual terms - i.e. - 'hammersandnails' as opposed to 'hammers and nails'.

Excessive punctuation in the TITLE and description tags – wastes precious space, and some characters are ignored or may cause a problem with the spider .

13. Conclusion: As I mentioned at the beginning, the semantic web is the idea of teaching the web -- which is to say teaching the next generation of web search engines and web browsers -- how to understand the content rather than just the structure on the web. It is teaching the engines to read, understand, draw out the essence and be able to deliver it to us. A primary use of this would be better search engines, but it also has other uses. Web browsers and search engines would be able to answer simple questions by mining the web for the answers. You might also be able to highlight a phrase in your browser and have it come up with more information about that phrase -- such as reading an article about horses, highlighting a phrase about the proper care for a horse, and having the browser pull up more information about that subtopic. But it is still a long road before we get there, and it is a road with many different paths.