International Journal of Scientific Research Engineering & Technology (IJSRET)
Volume 1 Issue5 pp 199-206 August 2012 www.ijsret.org ISSN 2278 – 0882

A Conceptual Study on Semantic Information Retrieval

[1]Rajat Goel, [2]Rajeev Kumar, [3]Shalu Goel, [4]Sudarshan Goswami
[1,3]Department of Computer Science & Engineering, Translam Institute of Technology & Management, Meerut
[2]M.Tech Student, Department of Computer Science & Engineering, IIMT Engineering College, Meerut
[4]Associate Professor, Department of Information Technology, IIMT Engineering College, Meerut
dr.rajatgoel@gmail.com, rajeev.msys@gmail.com, goelsha@gmail.com, goswami.sudarshan@gmail.com

ABSTRACT
The amount of content stored and shared on the Web and other document repositories keeps increasing steadily and fast. This growth results in well known difficulties and problems when it comes to finding and properly managing information in massive volumes. However, users still miss or need considerable effort sometimes to reach their targets, even if the sought information is present in the search space. A common cause for this is that currently consolidated content description and query processing techniques for Information Retrieval (IR) are based on keywords, and therefore provide limited capabilities to grasp and exploit the conceptualizations involved in user needs and content meanings. Aiming to solve the limitations of keyword-based models, the idea of conceptual search, understood as searching by meanings rather than literal strings, has been the focus of a wide body of research in the IR field.

Keywords**:** *Semantic Search, Keyword based search, Context Extraction, Semantic Annotation, Ontology, OWL.*

I. INTRODUCTION
A conceptual study on improving the searching techniques used by the semantic search engines keeping time complexity as the major factor. The documents available on the Web are poorly structured or unstructured. These documents once processed for semantic search has two parts- semantic annotated part and content rich text. The searching process requires querying both these parts with different query languages which are incompatible with each other. This adds as another reason for the slow development of Semantic Search Engine. The process of Semantic Web Search can be summarized in the following figure.

The idea of supporting a higher-level conceptual (computerized) understanding of contents and queries has been present in the IR field since the early eighties (Croft, 1986), if not earlier (Van Rijsbergen, 1979). More recently, it can be said to have become one of the "philosopher's stones" in the Semantic Web (SW) community since its emergence in the late nineties.

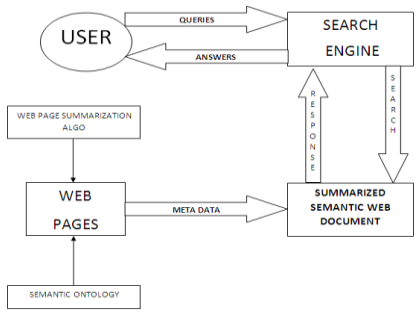

Figure 1

II. CONCEPT OF INFORMATION RETRIEVAL SCHEMA
The goal of an IR system [1] can be described as the representation, storage, organization of, and access to information items. This section provides a brief description of the different resources, components and tasks involved in an information retrieval system. The IR schema aims to introduce the main components that are developed in our semantic retrieval model.

Input: An IR system takes two main inputs, the user needs and the information items.

• *User needs:* An information retrieval schema [2] begins when a user expresses his information need to the system. In the general case, this information need is conveyed in the form of a search string, but it can also be expressed in other formats, as in the case of Multimedia Retrieval, where the input can be an image, sound, etc.

• *Information items:* Orthogonal to the kind of queries that can be asked is the subject of the information items the system adopts. The information item is the basic element which can be retrieved as an answer to a query and it is mainly classified by its format (textual

document, audio, video, image, etc) and its granularity (Web page, paragraph, sentence, etc).

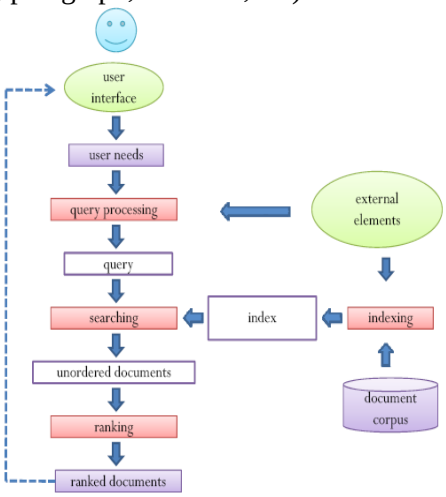

Fig 2: The Information Retrieval Schema

Output: And IR system typically returns one main output, consisting of a ranked list of information items.

• *Ranked information items:* This output consists of a sorted list of information items. The retrieved items may have different format (text, audio, video, etc) and structure. Regarding the structure, a large classification can be made distinguishing systems that return unstructured information (items with arbitrary structure and syntax, such as free text documents), and those that return specific structured information (such as relational databases objects). The systems that return structured information are commonly characterized as data retrieval systems. While these models do cannot deal with general information about the subject or topics involved in the sought data, they return very precise answers in response to specific, unambiguous, and formally expressed information needs.

Processes: Three main processes can be identified in an IR system: a) extraction of item content features and descriptors into a logic representation of items (indexing); b) handling user's information needs into an abstract representation (query processing) and, c) matching both representations (searching and ranking).

• *Indexing:* Not all the pieces of an information item are equally significant for representing its meaning. In written language, for example, some words carry more meaning than others.

Therefore, it is usually considered worthwhile to pre-process the information items to select the elements to be used as index objects. Indices are data structures constructed to speed up search. It is worthwhile building and maintaining an index when the item collection is large and semi-static. The most common indexing structure for text retrieval is the inverted file.

This structure is composed of two elements: the vocabulary and the term occurrences. The vocabulary is the set of all words in the text. For each word in the vocabulary a list of all the text positions where the word appears is stored. The set of all those lists is called occurrences.

• *Query processing:* The user needs, the query, are parsed and compiled into an internal form. In the case of textual retrieval, query terms are generally pre-processed by the same algorithms used to select the index objects. Additional query processing (e.g., query expansion)[5] requires the use of external resources such as thesauri or taxonomies.

• *Searching:* User queries are matched against information items. As a result of this operation, a set of potential information items is returned in response to user needs. The Information Retrieval achieved may vary considerably depending on the format of information (text, audio, video, etc), but in all cases, some form of simplification is done in the information model to make it tractable. For instance, text retrieval commonly builds on the assumption that the matching between information items (the documents) and user information needs (the query string) can be based on a set of index terms. This obviously involves an (acceptable –because reasonably effective– but considerable) loss of semantic information when text is replaced by a set of words. A similar situation occurs in multimedia retrieval where matching is performed based on numeric signal features.

• *Ranking:* The set of information items returned by the matching step generally constitutes an inexact, by nature approximate answer to the information need. Not all the items contain relevant information to the user. The ranking step aims to predict which how relevant the items are comparatively to each other, thus returning them by decreasing order of estimated relevance [12]. Thus, in a way, ranking algorithms can be considered the core of IR systems, as they are keys to determine the performance of the system.

External elements: External elements are mainly used in helping to represent, extract and process user needs and content meanings. The understanding of the semantics behind information items and users queries

helps to enhance the precision of the retrieval process, and therefore, to increase user satisfaction. Three main external elements are used within IR systems: a) the user interface, b) query processing operations and c) resources for indexing:

• *User Interface:* A flexible user interface is needed to allow the user to express his information needs but also to express possible constraints about the information he is looking for (e.g., exact content, similar content, disjoint content, content with a specific date, language, format, etc).

• *Query processing operations:* Depending on the type of query, different mechanisms can be used to refine it. The most common ones are based on additional user input. In this spectrum, relevance feedback approaches are generally the most efficient ones. However, they reduce the usability of the systems, and therefore other external resources, such as taxonomies and thesauri, are often used instead (or complementarily) to automatically classify, disambiguate or expand query terms.

• *Resources for indexing:* Document processing resources such as thesauri and controlled vocabularies can be used to help select the terms that are more appropriate as index objects.

III. METHODOLOGY

The major drawback of the keyword based search engine is its inability to analyze the relations between the keywords. As a solution to this idea of semantic web emerged but still is an unrealized dream due to the following reasons.

- Semantically annotating millions of documents available on Web is impractical.
- Since no one owns the Web; it is difficult to track whether new documents added to the Web are semantically annotated.
- Incompatibility between the query languages which are used to query the semantic annotation and text.
- The quality of the semantic annotations which is directly proportional to the relevancy of the results.
- Difficulty in processing RDF triplets if stored using tree/graph like data structure.

The semantic search engine proposed in this research works by resolving the above said issues. The first two problems can be solved by annotating the domains rather than annotating all the documents on the Web. This also keeps the Web designers free from the overhead of semantic annotation. It also paves the way for quality annotations as domains can be annotated by the experts. The processing of RDF triplets can be easier and fast by mapping the ontology to RDBMS. This also drastically reduces the time complexity of the searching algorithms.

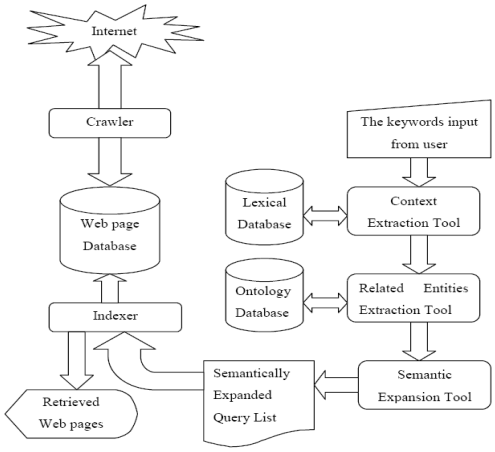

Fig 3: The Working Model of Methodology in IR System

(1) Context Extraction Phase: A word in English language can be used in different contexts depending on the way in which it is used (Noun, Verb, Adjective). Lexical semantics is the study of how and what the words of a language denote. Words may either be taken to denote things in the world, or concepts, depending on the particular approach to lexical semantics. Lexical semantics covers theories of the classification and decomposition of word meaning, the differences and similarities in lexical semantic structure between different languages, and the relationship of word meaning to sentence meaning and syntax. Meaning of a lexical unit is established by looking at its neighborhood, or if the meaning is already locally contained in the lexical unit or by mapping words to concepts.

The Context Extraction Phase requires the lexical database which organizes the information of lexicons and their semantic structure. A lexical database is a lexical resource which has an associated software environment database which permits access to its contents. The database may be custom-designed for the lexical information or a general-purpose database into which lexical information has been entered. Information typically stored in a lexical database includes lexical category and synonyms of words, as well as semantic relations between different words or sets of words.

Lexical database is chosen as the tool for the Context Extraction Phase because of the following reasons.

- Sense disambiguation is crucial for Information Retrieval [6].
- It is the best available resource for extracting the contexts of keywords.
- The database can be modified to include the new terms or relations whenever required.
- Ease of use and understandability.

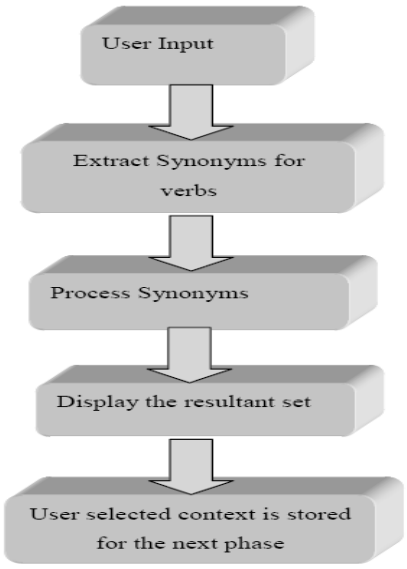

Fig 4: Working Model of Context Extraction Phase

Word Net Database:

Word Net, a lexical database developed by George. A Miller contains 155, 000 noun word forms organized into 112,000 lexicalized concepts. This database tries to make semantic relations between word senses more explicit and easier to use. But it omits pronunciations, derivative morphology, etymology, usage notes or pictorial illustrations; thus differs from the conventional thesaurus. The basic semantic relations between word forms in Word Net are synonymy which forms the building blocks. A synset in Word Net is a collection of synonyms where synonyms are other word forms which can be substituted for the given word without any change in the meaning.

Algorithm for context extraction phase

Algorithm 1 ContExt(Keyword k):
Input: The keyword k
Output: The selected context related to the keyword: SC.
(1) Set k as the keyword entered by the user.
(2) Generate IndexWord for keyword k as NOUN.
(3) Process IndexWord to generate synsets and store in IN_D document set.
(4) Display IN_D to the user
(5) Store the user selected one in SC.

(2) Related Entities Extraction Phase

Related Entities Extraction (REE) is the second phase in the model proposed. It works on extracting the entities relating to the context of the keyword selected by the user in the previous phase. The input of this phase is the output of previous phase and output of this phase is semantically expanded query list which will be fed to the next phase as input.

Pre-requisite and tools used

1. Pre-categorized documents which are classified using text categorization methods which employs semantic methods.
2. A very well semantically annotated domains
3. Ontology database mapped on to RDBMS

Semantic Annotation

Semantic web is about adding formal semantics (metadata, knowledge) to the web content for the purpose of more efficient access and management. Adding of metadata and knowledge and knowledge to a document which specifies what the document contains is known as semantic annotation. Semantic annotations in a document are additional information that identifies or defines a concept in semantic model in order to describe a part of the document. RDF (Resource Description Framework) offers a framework to model hierarchies of classes and properties. RDF is designed for specific data about specific subjects. RDF can represent data and metadata that describes the data very well.

In RDF, a statement links two resources. The statement is viewed as sentences that have subject-verb-object structure. Hence it is called as a triple. The subject of the statement is in fact called subject. The equivalent part of the verb is predicate and the remaining part is called object.

Eg. I	live	in	Bangalore
(subject)	(predicate)		(object)

Algorithm of Related Entities Extraction Phase

Algorithm: RelEntExt(String Context, String Keyword)
Input: Keyword and the context selected by the user.
Output: RE_S, A set of related entities extracted.

Steps:
1. Set O_S= Φ.
2. For every domain Di in D_S,
    I. Retrieve subjects where object is the keyword and
        predicate is 'aka' and add these to semantic query list
        RE_S.
    II. Retrieve subjects where object is the keyword and
        predicates are instance of' and apart of'. Add these
        to O_S temporary output list.
    III. Retrieve objects where subject is the keyword and
        Predicates are _a kind of'. Add these to O_S temporary output list.
3. Output O_S to the user.
4. If
    I. The user is satisfied go to Step # 5
Else
    I. Read the new keyword set K= new Keyword.
    II. Call RelEntExt(Context, K).
5. Output RE_S.
6. Stop.

(3) Searching the Web Phase

The third phase is responsible for searching in the heterogeneous pool of web documents. This phase input is not just a keyword but a set of related terms connected by Boolean operators. The output of this phase is the net output for the whole search process which will be displayed to the user. The tools used are Google Advanced Search Interface, Indexer and crawler used by search engines. The Basic search of Google help to covers all the most common issues, but sometimes when there is need of more specific power this search fails. Google has developed Google Advanced Search keeping this scenario in mind. It improvises the way of searching by searching the exact terms omitting certain user specified words. It works by implementing the following methods.

- **Phrase search ("")**

By putting double quotes around a set of words, Google considers the exact words in that exact order without any change.

- **Search within a specific website (site :)**

Google confines the search to user specified websites so that your search results must come from a given website. You can also specify a whole class of sites.

- **Terms to exclude (-)**

Attaching a minus sign immediately before a word indicates that we do not want pages that contain this word to appear in your results. The minus sign should appear immediately before the word and should be preceded with a space. For example, in the query [anti-virus software], the minus sign is used as a hyphen and will not be interpreted as an exclusion symbol; whereas the query [anti-virus -software] will search for the words 'anti-virus' but exclude references to software. You can exclude as many words as you want by using the - sign in front of all of them, for example [jaguar -cars -football -os]. The - sign can be used to exclude more than just words. For example, place a hyphen before the 'site:' operator (without a space) to exclude a specific site from your search results.

- **Fill in the blanks (*)**

The *, or wildcard, is a very powerful that tells Google to try to treat the star as a placeholder for any unknown term(s) and then find the best matches. For example, the search [Google *] will give you results about many of Google's products (go to next page and next page -- we have many products). The query [Obama voted * on the * bill] will give you stories about different votes on different bills. Note that the * operator works only on whole words, not parts of words.

- **Search exactly as is (+)**

If a + is attached immediately before a word Google matches that word precisely as typed by the user. Putting double quotes around a single word will create the same impact.

- **The OR operator**

If you want to specifically allow either one of several words, you can use the OR operator. For example, [San Francisco Giants 2004 OR 2005] will give results about either one of these years, whereas [San Francisco Giants 2004 2005] (without the OR) will show pages that include both years on the same page. The symbol | can be substituted for OR.

- **The AND operator**

If you want to insist on having all the keywords, we can use the AND operator. By default, Google Advanced Search combines each and every keyword with AND..
The symbol & can be substituted for AND.

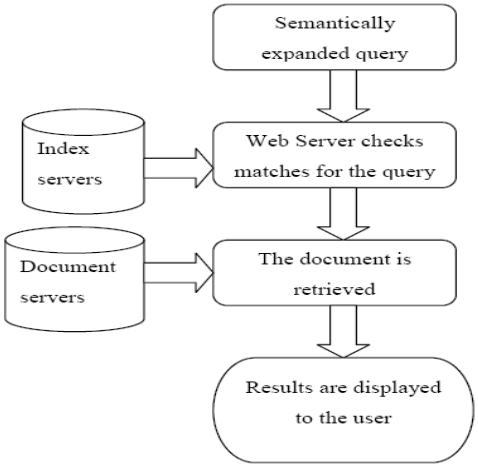

Fig 5: The Working Model of Searching the Web Phase

Google runs on a distributed network of thousands of low-cost computers and can therefore carry out fast parallel processing. Parallel processing is a method of computation in which many calculations can be performed simultaneously, significantly speeding up data processing. Google has three distinct parts:
1. Googlebot, a web crawler that finds and fetches web pages.
2. The indexer that sorts every word on every page and stores the resulting index of words in a huge database.
3. The query processor, which compares your search query to the index and recommends the documents that it considers most relevant.

IV. EXPERIMENTAL RESULTS

A web is the place where documents are available for download on the internet. Imagine if there would be no hyperlinks among them. You would not be able to navigate among the web pages without the hyperlinks. As we know that the data on the web is not enough for the increasing users of internet globally. So we need a proper infrastructure for a real web of data.

- The data available on the web must be accessible via standard Web technologies.
- The data should be interlinked over the Web i.e., the data can be integrated over the web.

SPARQL query engine is used for querying purpose. It is a semantic web tool for querying on RDF graph structures. We used Jena toolkit [7] in Java to build the query interface for the user through SPARQL.

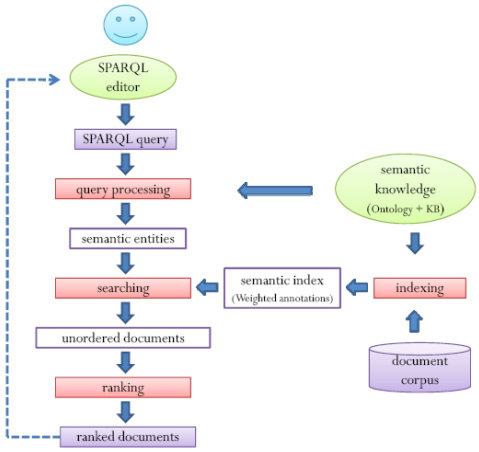

Fig 6: A Graphical Representation of Semantic Information Retrieval

As we can see in the figure, this ontology-based IR model [11] is an adaptation of the classic keyword based IR model. It includes its four main processes: indexing, querying, searching and ranking. However, as opposed to traditional keyword-based IR models, in this approach the query is expressed in terms of an ontology-based query language (SPARQL) and the external resources used for indexing and query processing are ontology and its corresponding KB. The indexing process is equivalent to a semantic annotation process. Instead of creating an inverted index where the keywords are associated with the documents where they appear, in the case of our ontology-based IR model, the inverted index contains semantic entities (meanings) associate to the documents where they appear. The relation or association between a semantic entity and a document is what we call annotation.

The overall retrieval process is illustrated and consists of the following steps:

- Our system takes as input a formal SPARQL query.
- The SPARQL query is executed against a KB, which returns a list of semantic entities that satisfy it. This step of the process is purely Boolean (i.e. based on an exact match), so that the returned instances must strictly hold all the conditions in the formal query.
- The documents that are annotated (indexed) with these instances are retrieved, ranked, and presented to the user. In contrast to the previous phase, the document retrieval phase is based on

an approximate match, since the relation between a document and the concepts that annotate it has an inherent degree of fuzziness.

Result of Context Extraction Phase:

The Context Extraction Phase of this model retrieves the different contexts in which a word can appear. The lexical database Word Net is used along with JWNL functions which are plugins for Java developed by Princeton University in order to use this database. For analysis purpose, the keyword Cancer is selected.

For instance, the following are the contexts that are retrieved for the keyword "Cancer" a noun from Word Net by Context Extraction Phase developed using JWNL functions.

C:\Program Files\jwnl14-rc2>java jwnltrial1 Cancer
Oct 4, 2010 9:49:16 PM net.didion.jwnl.util.MessageLog doLog
INFO : Installing dictionary net.didion.jwnl.dictionary.FileBackedDictionary@18e3e60

1: any malignant growth or tumor caused by abnormal and uncontrolled cell division; it may spread to other parts of the body through the lymphatic system or the blood stream
2: (astrology) a person who is born while the sun is in Cancer
3: a small zodiacal constellation in the northern hemisphere; between Leo and Gemini.

Result of Related Entities Extraction Phase:

The Context Extraction Phase result is the context selected for the next phase and in this case it is "Disease". Proceeding with this in the current phase yields the following result. The output of this phase is semantically expanded query combined with OR and AND operators.

(Blood cancer OR haematological Malignancy OR Leukemia OR Multiple Myeloma) AND (Cancer OR Malignant Neoplasm OR oncology)

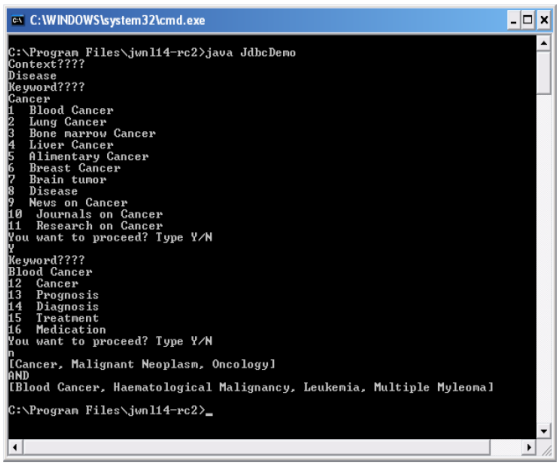

Figure 7

Result of Searching the Web Phase:

The output of the previous phase is given as the query phrase in Google Advanced Search in this phase. The output is as follows.

Figure 8

V. CONCLUSION

Semantic Web is considered as Web of Data. It is not the newer version of Web but it only advocates for the conversion of existing contents of Web into machine readable form. The machines require semantics information to establish relationship among the content. The major limitation of current search engines is the lack of these missing semantics in current Web contents. This results in huge number of retrieval of results. Most of them are neither reliable nor relevant.

The major drawback of the keyword based search engine is its inability to analyze the relations between the keywords. Semantic search engines were evolved an as highly demanded solution for this problem. Semantic Search Engines is still an unrealized dream due to the various reasons such as difficulty in annotating millions of documents available on Web, it is difficult to track whether new documents added to the Web are semantically annotated, Incompatibility between the query languages which are used to query the semantic annotation and text, The quality of the semantic annotations which is directly proportional to the relevancy of the results, Difficulty in processing RDF triplets if stored using tree/graph like data structure.

REFERENCES

[1] Artem Chebotko, Shiyong Lu, and Farshad Fotouhi, Semantics Preserving SPARQL-to-SQL Translation, Data & Knowledge Engineering 2009.

[2] Hakia Team, hakia Semantic Search Technology – making sense of the worlds information. White paper Jan 2010.

[3] D.Allemang and J.Hendler: Semantic Web for the Working Ontologist,2008.

[4] Lizhen Li, Zhifeng Dong, Keming Xie, *Ontology of general concept for SemanticSearching*, Second International Conference on Computer Modeling and Simulation 2010.

[5] Agosti, M., Crestani, F., Gradenigo, G., & Mattiello, P. (1990). An approach to conceptual modelling of IR auxiliary data. IEEE International Conference on Computer and Communications, Scottsdale, Arizona.

[6] Baeza Yates, R., & Ribeiro Neto, B. (1999). Modern Information Retrieval. Harlow, UK: Addison-Wesley

[7]Y. Bitirim, Y. Tonta, H. Sever, "Information Retrieval Effectiveness of Turkish Search Engines. In Advances in Information Systems", Lecture Notes in Computer Science, T. Yakhno (Ed.), vol. 2457, pp. 93-103, Springer-Verlag, Heidelberg, 2002.

[8] D. Tümer, M. Ahmed Salah, Y. Bitirim, "An Empirical Evaluation on Semantic Search Performance of Keyword-Based and Semantic Search Engines: Google, Yahoo, Msn and Hakia", In Fourth International Conference on Internet Monitoring and Protection. IEEE press, Venice, Italy, 2009.

[9] Jae Hyun Lim, Young-Chan Kim, Hyonwoo Seung, Jun Hwang , Heung-Nam Kim, "Query Expansion for Intelligent Information Retrieval on Internet", Proceedings of the International Conference on Parallel and Distributed Systems, Page(s):652 - 656, 1997

[10] O. Corcho, M. Fernández-López, A. Gómez-Pérez, and A. López-Cima, Building legal ontologies with METHONTOLOGY and WebODE. *Law and the Semantic Web*, No., pp. 142-157, 2003.

[11] B. McBride, Jena: A semantic web toolkit. *IEEE Internet Computing*, Vol. 6, No. 6, pp. 55-59, 2002.

[12] Mayfieldeld, J., Finin, T.*: Information Retrieval on the Semantic Web: Integrating Inference and Retrieval.* in: Proc. of the Int'l Workshop on the Semantic Web at the 26th Int'l ACM SIGIR Conf.e on Research and Development in Information Retrieval, Toronto, Canada (2003).

[13] Berners-Lee, Tim; James Hendler and Ora Lassila (May 17, 2001). "The Semantic Web". *Scientific American Magazine*. URL: http://www.sciam.com/article.cfm?id=thesemantic-web&print=true.

[14] Breitman, K. K., Casanova, M. A., & Truszkowski, W. (2007). *Semantic Web: Concepts, Technologies and Applications.* London, UK: Springer-Verlag. E. Prud'Hommeaux and A. Seaborne, SPARQL query language for RDF. *W3C working draft*, Vol. 20, No., 2006.

[15] Buckley, C., & Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. *27th annual international ACM SIGIR conference on Research and development in information retrieval*, (pages 25 -32). Sheffield, United Kingdom.

[16] Cooper, W.S.: *A Definition of Relevance for Information Retrieval.* in: J. Information Storage and Retrieval, 7(1), pp. 19-37 (1971)