A Study on Searching Mechanisms in Semantic Web

M.Thangaraj[1] and G.Sujatha[2]

[1]Madurai Kamaraj University, Department of Computer Science,*thangarajmku@yahoo.com*
[2]Sri Meenakhsi Govt. College For Women (A), Department of Computer Science.

[2]Corresponding Author *sujisekar05@rediffmail.com*

Abstract: The internet presents a huge amount of useful information which is usually formatted for its users which makes it difficult to retrieve relevant data from various sources. With the growing complexity of online information the search results in keyword based search engine are growing increasingly vague and cumbersome. The existing information retrieval systems are mostly keyword-based and retrieve relevant documents or information by matching keywords. Keyword-based search in spite of its merits of expedient query for information and ease-of-use has failed to represent the complete semantics contained in the context and has led to the retrieval failure. Although many approaches for Information Retrieval in semantic web has been developed, there has been limited effort to compare such tools. The architectural aspects of a few semantic search systems were presented by comparing various features.

Keywords: Semantic Web, Semantic search, Information Retrieval, Ontology, Search Engine, Semantic Similarity.

1. Introduction

The semantic web [6] is an extension of the current Web in which resources are described using logic-based knowledge representation languages for automated machine processing across heterogeneous systems. In recent years, its related technologies have been adopted to develop semantic-enhanced search systems.

Semantic Search Systems (SSS) are Information Retrieval (IR) Systems that employ semantic technologies to enhance different parts of IR by using semantic Relations, Ontologies, Clusters, Crawlers and Similarities. Research in IR community has developed variety of techniques to help people locate relevant information in large document repositories.

Besides classical IR models i.e., Vector Space and Probabilistic Models[4] extended models such as Latent Semantic Indexing, Machine Learning based models i.e., Neural Network, Symbolic Learning, and Genetic Algorithm based models and Probabilistic Latent Semantic Analysis (PLSA) have been devised with hope to improve information retrieval process. However, rapid expansion of the Web and growing wealth of information pose increasing difficulties to retrieve information efficiently on the Web. To arrange more relevant results on top of the retrieved sets, most of contemporary Web search engines utilise various ranking algorithms such as PageRank, HITS, and Citation Indexing that exploit link structures to rank the search results. Despite the substantial success, those search engines face perplexity in certain situations due to the information overload problem on one hand, and superficial understanding of user queries and documents on the other.

Significance of the research in this area is for two reasons: it supplements conventional information retrieval by providing search services centered on entities, relations, and knowledge; and development of the semantic web also demands enhanced search paradigms in order to facilitate acquisition, processing, storage, and retrieval of the semantic information. This paper provides a survey to gain an overall view of the current research status in this area. We classify our studied systems into several categories according to their most distinctive features, as discussed in the next section. The categorization by no means prevents a system from being classified into other categories. We provide a review focusing on objectives, methodologies, and most distinctive features of individual systems; and discuss issues related to knowledge acquisition and search methodologies.

In this paper, We focus on Semantic Search architectures from five directions. They are i. Relation Centered Search ii . Ontology Centered Search iii. Similarity Based Search iv.Crawler Based Search v. Cluster Based Search. This paper is organized as follows. Section 2 introduces related work in this area. Then the semantic search directions are presented in Section 3. Finally the conclusions are made in Section 4.

2. Semantic Search Systems

The unsolved problems of current search engines have led to the development of the semantic web search systems [31]. Search is one of the most popular applications on the web and it is an application with significant room for improvement. The addition of explicit semantics can improve search. Semantic search attempts to augment and improve traditional search results by using data from the semantic web [10].

Variety of SSS consists of different tools: semantic browsing with automatically generated annotations, Semantic Query expansion, Semantic Ranking, Systems working on a Single Ontology or Multiple Ontologies. There exist various attempts to classify the searching system. For instance, distinguish four key characteristics of semantic metadata based search systems: search environment, query type, intrinsic problems, iterative and exploratory dimensions.

Furthermore, the SSS are classified by semantic technology usage, and the usage of ontology and its elements. We summarize important categories [7] in Fig. 1 based on analysis of the literature and related classification schemes.

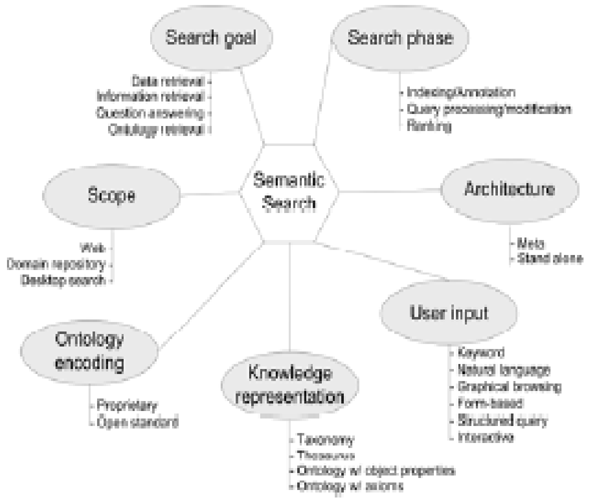

Fig 1: Classification of semantic search systems

3. Directions in Semantic Search System

The Semantic Search architectures from five directions i.Relation Centered Search ii. Ontology Centered Search iii.Similarity Based Search iv.Crawler Based Search v. Cluster Based Search are discussed in this section.

3.1 Relation-Centered Search

Relation-Based search is an extension of the conventional IR approaches where the main goal is to retrieve the most meaningful pages only. In this type of search system the retrieval process is carried out by matching user queries with the relationship between keywords.

The first approach for ranking the pages is based on the content count for a particular keyword [3]. For the given noun N, the frequency of occurrence of N in the database is keyword searching, finding the frequency of occurrence for its relations in that page gives us a count of how meaningful the page is towards N. The page which has a highest number of related keyword hits a higher page rank for the user entered keyword. Consider the three web documents, the parts of speech, such as nouns, verbs and adjectives were extracted using a grammatical parser like Link Grammar Parser. These parts were then fed to a lexical dictionary WordNet, to extract the various relations such as synonyms, hypernyms, hyponyms, meronyms, and holonyms. These relations were then stored in tables categorized by their part of speech, for e.g. P1N, P2N, P3N, P1ADJ, P2ADJ, P3ADJ, P1V, P2V, and P3V, where P1N stands for "nouns on page1 with its extracted relations", P1V stands for "verbs on page1 with its extracted relations", and so on. Now the user query was processed by an algorithm depicted in Fig. 2.

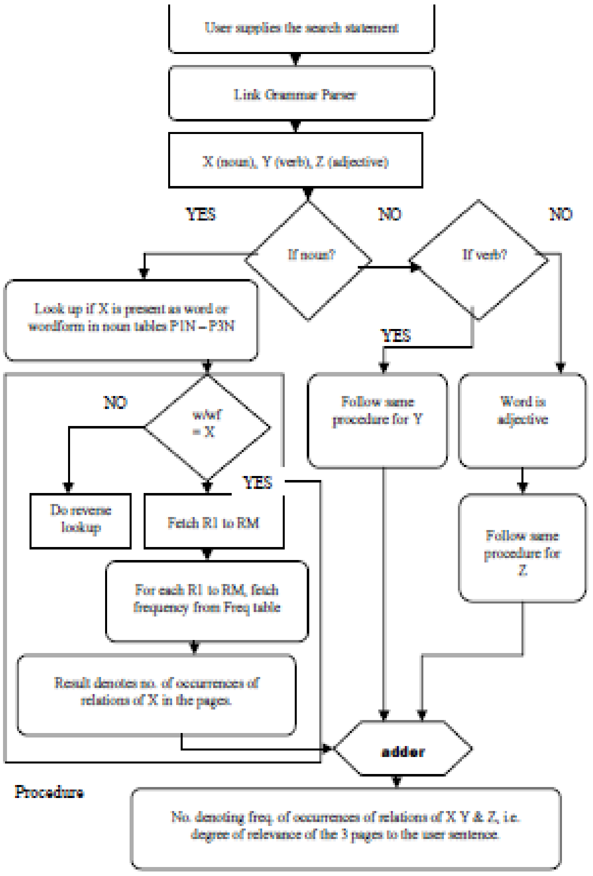
User supplies the search statement
Link Grammar Parser
X (noun), Y (verb), Z (adjective)
YES
NO
NO
If noun?
If verb?
Look up if X is present as word or wordform in noun tables P1N – P3N
YES
NO
w/wf = X
YES
Do reverse lookup
Fetch R1 to RM
For each R1 to RM, fetch frequency from Freq table
Result denotes no. of occurrences of relations of X in the pages
Follow same procedure for Y
Word is adjective
Follow same procedure for Z
adder
Procedure
No. denoting freq. of occurrences of relations of X Y & Z, i.e. degree of relevance of the 3 pages to the user sentence.

Fig 2. Algorithm to Perform Related Keyword Search

The Nouns, Verbs, Adjectives (X, Y, Z) were first extracted from the user queried sentence. The noun X, was first checked in the three tables of nouns for the three web pages. If X was present physically in any of the three web pages, would get a hit in the "word" column or "word form" column of the three noun tables P1N-P3N. If there is a hit, consider the corresponding relations for noun X. Then perform a frequency search for those relations of X in the three pages. So in this way, the system searched for related keywords or keyword relations in a webpage. This count gave an idea of how relevant the webpage was towards the keyword. Similar treatment was allotted to the other keywords of the user queried sentence. The total count got was a measure of the page relevance towards the user queried sentence. This work is based on the relation count between the keywords. It was improved by adding semantic meanings in the SemSearch System.

"SemSearch" hides the complexity of semantic search from end users and to make it easy to use and effective for novice users [33]. SemSearch is a layered architecture (Fig.3) that separates end users from the back-end heterogeneous semantic data repositories. User Interface Layer, allows end users to specify queries in terms of keywords. The Text Search Layer makes sense of user queries by finding out the explicit semantic meanings of the user keywords. Two components namely a semantic entity index engine (indexes documents and their associated semantic entities including classes and properties) and semantic entity search engine (supports the searching of semantic entity matches for the user keywords) are central to this layer.

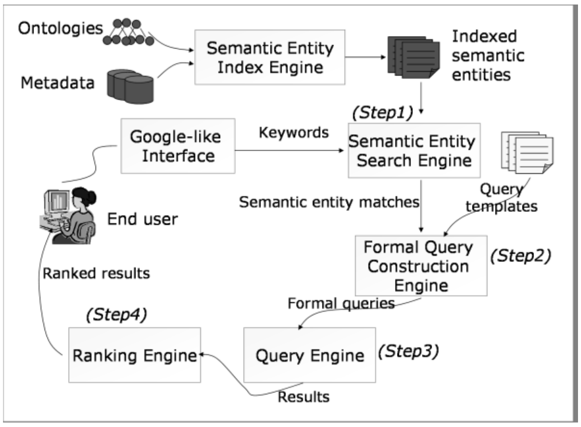

Fig. 3. An overall diagram of the SemSearch search engine

The semantic query layer produces search results for user queries by translating user queries into formal queries which comprises of a formal query construction engine (provides a specific formal query language that can be used to retrieve semantic relations from the underlying semantic data layer), a query engine (queries the specified metadata repository using the generated formal queries) and a ranking engine (ranks the search results according to the degree of satisfaction on the user query). The semantic data layer, comprises of semantic metadata that are gathered from heterogeneous data sources and are represented in different ontologies. SemSearch accepts keywords as input and produces results which are closely related to the user keywords in terms of semantic relations. This method is modified by formulating concept based keywords.

The early work [34] "Ontolook", is a relation based search engine provides the relationship between the keywords in terms of the concepts. Initially "OntoLook" will analyze the keyword combination input by the user. The system will analyze these inputs and handle the inherited relationship between these concepts. Then, these concepts are assembled to some concept pairs and send these pairs to the ontology database to retrieve all relations defined by ontology between concept pairs.

After all relations between concept pairs are retrieved from the ontology database, a concept-relation graph is formed based on these relations and concepts. Then "OntoLook" will cut less relevant arcs from the graph. If the number near the arc is larger then it denotes the maximum relations between the concepts. Otherwise if the number near the arc is zero, the algorithm behaves like a Keyword based search. Because there are some relations between the keywords which user input, the result set retrieved from the database will be close to the users' intention when less-ranked arcs were cut from the graph. Finally, the system fetches the relation and its corresponding keyword pair from each arc in sub graphs to form a property-keyword candidate set. Then, the property keyword candidate set is sent to the database to get a retrieved result set for the users.

In this architecture (Fig 4) a crawler program collects the web pages on the internet with its semantic markup and corresponding ontology which is described in an OWL document in the Internet. The collected web pages are transported to a web page database to be stored for the use of future retrieving URLs and corresponding web pages. The ontology, OWL document, is conveyed to an OWL Parser. The OWL parser will map the ontology into a relational database.

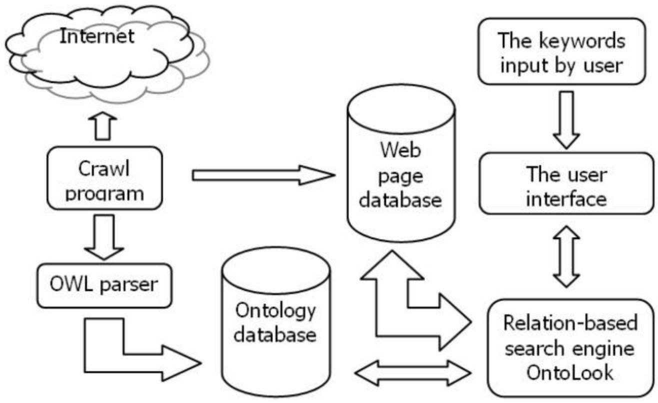

Fig. 4. System architecture of "OntoLook.

The effectiveness of this approach is limited by lack of priority ranking technology and page rank technology to make a relation based page rank. With this considerations the next work is presented in [9] by providing effective page ranking. The Annotated Web pages from the SemanticWeb including RDF metadata are collected by the crawler application and originating OWL ontology. The OWL Parser interprets the RDF metadata and stored in the knowledge database. A graphics user interface allows for the definition of a query, which is passed on to the relation-based search logic. The ordered result set generated by this latter module is finally presented to the user.

The "ranking criterion" (Fig 5) is based on the estimate of the probability that keywords/concepts within an annotated page are linked one to the other in a way that is the same to the one in the user's mind at the time of query definition. This probability measure can be effectively computed by defining a graph-based description of the ontology (ontology graph), of the user query (query subgraph), and of each annotated page containing queried concepts/keywords (both in terms of annotation graph and page subgraph).

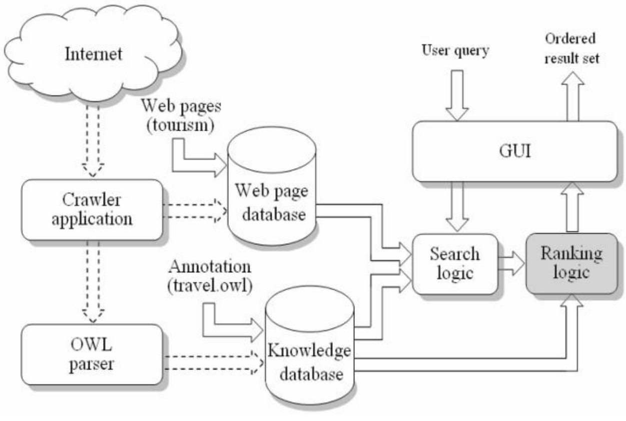

Fig 5. Semantic Web infrastructure (prototype architecture).

Given an ontology graph and a query subgraph a ranking strategy is designed. This strategy will assign a relevance score to each page including queried concepts based on the semantic relations. As per the proposed ranking strategy, for the given query Q, for each page p, a page subgraph can be built and exploiting the information available in page annotation. The methodology starts from a page subgraph computed over an annotated page and generates all the possible combinations of the edges belonging to the subgraph by excluding cycles.

There may be pages in which there are concepts that do not show any relations with other concepts. But that could still be of interest to the user. The methodology progressively reduces the number of edges in the page subgraph. Then it computes the probability of the resulting subgraphs obtained by a combination of the remaining edges that matches the user's intention. Edge removal could lead to having concepts without any relation with other concepts. Thus, several relevance classes are defined, each characterized by a certain number of connected concepts. Within each class, pages are ordered depending on the probability measure above and presented to the user. An enhancement of present work with multiple ontologies is seemed to be effective.

3.2 Ontology-Centered Search

Ontology provides a flexible way of introducing semantics into the semantic web. The main advantage of using ontologies is reusability of knowledge. A number of ontology libraries currently exist. Example libraries are Ontolingua (www.ksl.stanfor.edu/software/ontolingua) and OWL library (http://protege.stanford.edu/plugins/owl/owl-library). To get the right information, the search engines must be capable of finding the suitable ontologies. Some ontology based search engines available currently are Swoogle, Ontosearch.

In Ontosearch [32] which combines the google search engine together with the RDFs ontology (hierarchy) visualization technology. It will search for relevant (based on keywords) ontology files on the Internet and displays the files in a visually appealing way—as a hierarchy tree. The hierarchical view allows the users to quickly review the structures of different ontology files and select the suitable ontology files.(Fig.6)

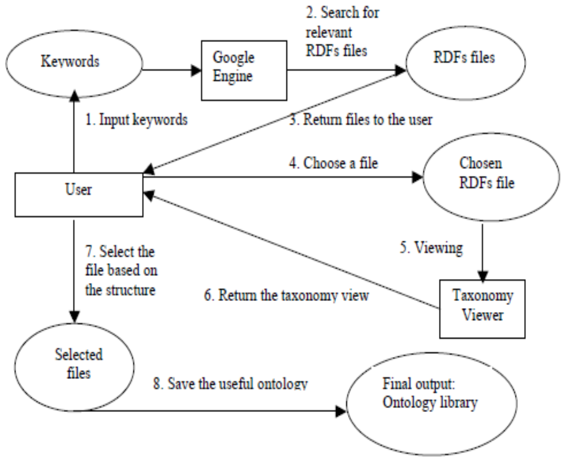
Keywords
Google Engine
2. Search for relevant RDFs files
RDFs files
1. Input keywords
3. Return files to the user
User
4. Choose a file
Chosen RDFs file
5. Viewing
7. Select the file based on the structure
6. Return the taxonomy view
Taxonomy Viewer
Selected files
8. Save the useful ontology
Final output: Ontology library

Fig 6. Overview of OntoSearch

The user inputs the keywords to describe the nature of the required ontology to OntoSearch. Then OntoSearch applies the Google engine to search for RDFs files related to the keywords and returns a list of relevant links (URLs) to the user. The user then chooses some of the returned RDFs files and displays their structure, and decides which of the files are relevant. Finally, the user select the relevant RDF files and saves them in a taxonomy library.

Ontology-searching tool OntoSearch, can be linked to the tool Information Knowledge Base (IKB). Fig 7 discusses links between them and demonstrates how they interoperate for future use.

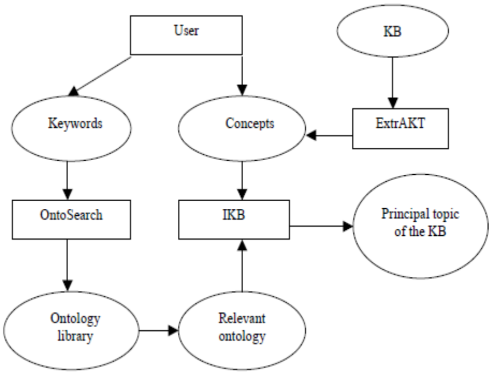

Fig 7. The relation between OntoSearch, IKB and ExtrAKT

The improved version of the previous work is based on Ontology-Based Knowledge Base by using vector space.

In the literature of semantic search engine based on Ontologies [28], the traditional Term Document Matrix (TDM) is extended to reflect the relevance between Ontologies, Web documents and terms. This extension of the traditional Vector Space Method (VSM) with semantic support. The search process begins with the parsing of a user's query (Fig 8). If a search request is in form of keyword list, then these keywords would be first treated as concepts in ontology, and documents that relates to these concepts will be retrieved based on the extended TDM. Through a user interface, a user can also submit requests by using a search wizard where user is given advanced options for a query. These options may include the ontology server, premises, answer patterns, maximum number of answers, and so on. In either forms, the request will be parsed into OWL-QL and then sent to the Reasoner which will return a set of RDF (Resource Description Framework) [36] triples containing qualified concepts/individuals in domain ontologies. After that, a document retriever finds all documents that are relevant to these concepts/individuals, and these documents are sorted by a ranker based on the relevance to the search request before they are presented to the user.

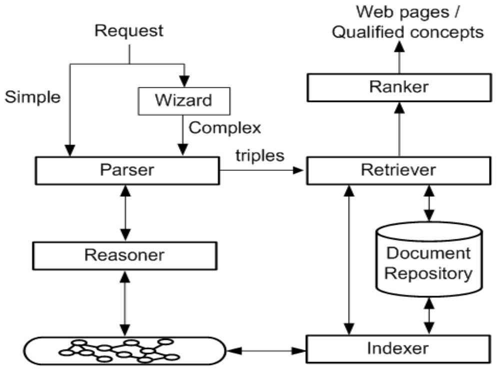

Fig 8. Query processing

The modification of the previous work is in [17] as the exploitation of ontology-based knowledge bases to improve search over large document repositories. This approach deals with an ontology-based scheme for the semiautomatic annotation of documents and a retrieval system. The retrieval model is based on an adaptation of the classic vector-space model, which includes an annotation weighting algorithm, and a ranking algorithm.

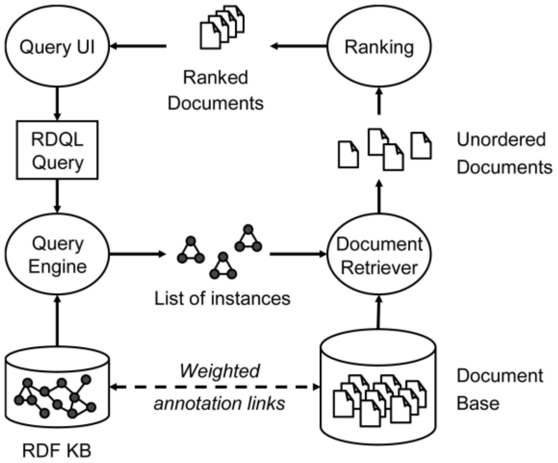

Fig 9. Overview of Ontology Based IR

This approach can be viewed as an evolution of classic keyword-based retrieval techniques, where the keyword-based index is replaced by a semantic knowledge base. The overall retrieval process is illustrated in Fig.9 that consists of the following steps: The input to the system is a formal RDQL query. The RDQL query is executed against the knowledge base, which returns a list of instance tuples that satisfy the query. 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. Finally, the documents that are annotated with the instances returned in the previous step are retrieved, ranked, and presented to the user. The efficiency of this method is improved by using inverted list indexing structure in Ontology Knowledge Bases.

The Ontology based Information Retrieval System uses inverted tables [35]. A new third layer in the existing 2-layer inverted list is introduced for storing the ontology terms belonging to the corresponding keywords. The architecture of the system consists of two parts: the information storage part (runs background and offline) and the query part (query runs instant and online). The ontology terms of query is corresponding to the terms in the inverted files, which could improve precision of the system.

The previous work is modified with the help of semantic annotations [12]. A semantic expansion search is proposed based on constructed domain ontology, semantic annotation algorithm and semantic expansion reasoning algorithm. The experimental results show that this methodology can overcome limitations in comparison with traditional keyword search mode, and achieve higher recall ratio and precision ratio.

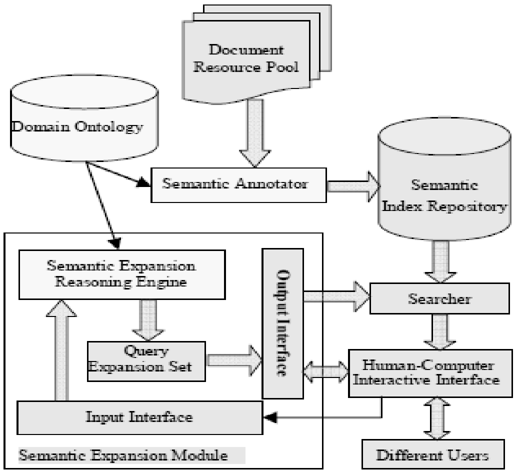

Fig.10. Semantic Expansion Search Model

Semantic expansion search model Sem-Exp-M is shown in Fig. 10. The function of semantic expansion module is to implement semantic expansion for user's query keyword. By the acquisition of search condition from human-computer interactive interface, reasoning engine executes reasoning and generates query expansion set via semantic expansion reasoning algorithm. Semantic annotator is to convert document resource pool with semantic feature. Searcher acquires query expansion set as search condition from output interface and retrieves documents from semantic index repository. This work is improved by introducing logic reasoned in the next model.

The logical reasoning based information retrieval model for the Semantic Web [24], uses OWL Lite as standard ontology language. The terms defined in ontology are used as metadata to markup the Web's content; these semantic markups are semantic index terms for information retrieval. The equivalent classes of semantic index terms by using description logic reasoner can be obtained. The logical views of documents and user information needs, generated in terms of the equivalent classes of semantic index terms. The performance of information retrieval can be improved effectively when suitable ranking function is chosen. Fig 11.

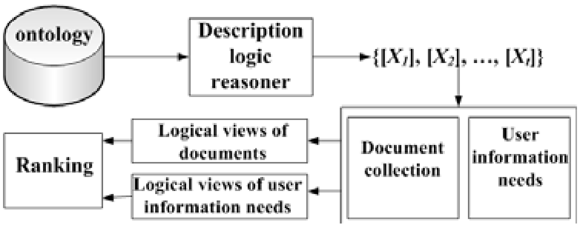
ontology
Description logic reasoner
{[X1], [X2], ..., [Xt]}
Logical views of documents
Ranking
Document collection
User information needs
Logical views of user information needs

Fig 11. Key parts of ontology-based information retrieval

The next work presents a methodology for the ontology based semantic annotation of web pages with annotation weighting scheme [25]. The retrieval model is based on the importance factors of the structural elements, which are used to re-rank the documents retrieval by the ontology based distance measure. The relevance concept similarities are combined with the annotation-weighting scheme to improve the relevance measures. A number of annotation tools for producing Semantic markups exist such as SHOE, Protégé, OntoAnnotate and MnM [11] [13].

The previous work is improved by adding ranking for ontologies. The study of "Ranking Ontologies Based on OWL Language Constructs" [20] uses more than one ontology to get the right kind of information the user is looking for. To present the suitable ontology to the user the ontologies are ranked by measuring two scores such as 1) How well the concept is described in terms of OWL constructs in a particular relevant class? 2) How much portion of the given ontology has the relevant OWL classes that describe the concept the user is looking for? Ontoweight is calculated by the Ontology Ranking Engine. The ontology that has the highest Ontoweight score will be ranked first.

3.3 Similarity Based Search

Similarity ranking is a hot topic in database research. Determining the semantic similarity is an important issue in the development of semantic search technology. An approach to determine the semantic similarity [15] between two entities that reflects in context. The semantic ranking approach assigns a value to the total number of entities and relations that match a user's interests.

The ranking score is defined as a function of some particular parameters. An Approach to Determine Semantic Similarity (ADSS) combines the Tabu Search algorithm with an efficient multiobjective programming algorithm to improve precision. Aleman-Meza et al [2] discuss a framework that uses ranking techniques to identify more interesting and more relevant semantic associations and define a ranking formula that considers subsumption weight, path length weight, and context weight and trust weight for assessing the effectiveness of the ranking scheme outlined. Rodriguez and Egenhofer [22] present an approach to computing semantic similarity across different ontologies.

A similarity function determines similar entity classes by using a matching process over synonym sets, semantic neighborhoods, and distinguishing features. In the SWAP project, Broekstra et al. [5] aim at overcoming the lack of semantics by combining the Peer-to-Peer paradigm with Semantic Web technologies. They propose a data model for encoding semantic information that combines ontology features with a flexible description and rating model. In Rodriguez and Egenhofer's approach, three ideas are presented—word matching, feature matching, and semantic-neighborhood matching. Broekstra et al. extend Rodriguez and Egenhofer's approach with a fourth idea—instance matching.

Thus, two objects can be identified through these similarity measures. Pekar and Staab [18] address the problem of automatically enriching a thesaurus by classifying new words into its classes. The proposed classification method uses the distributed data about a new word and the strength of the semantic relatedness of its target class to the other likely candidate classes. In contrast to the above work, ADSS introduces a multiobjective programming algorithm to compute the weights and the Tabu Search to compute the optimal solution. Hence the approach can acquire the results with higher precision. This method is modified with the heuristic mapping method in the next work.

One of the literature named "Ontology Mapping for information retrieval" [30] deals with a heuristic mapping method and a prototype mapping system that support semi-automatic ontology mapping for improving semantic interoperability in heterogeneous systems. This approach (Fig 12) is based on the idea of semantic enrichment, i.e., using instance information of the ontology to enrich the original ontology and calculate similarities between concepts in two ontologies. This approach consists of two phases: enrichment phase and mapping phase. The enrichment phase is based on analysis of the extension information in the ontologies.

The extension made in this work is written documents that are associated with the concepts in the ontologies. The intuition is that given two to-be-compared ontologies, construct representative feature vectors for each concept in the two ontologies. The documents are ''building material'' for the construction process, as they reflect the common understanding of the domain. Outputs of the enrichment phase are ontologies with feature vector as enrichment structure. The mapping phase takes the enriched ontology and computes similarity pair wise for the element in the two ontologies. The calculation is based on the distance of the feature vectors. Further refinements are employed to re-rank the result via the use of WordNet.

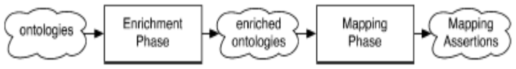

Fig. 12. Two phases of the whole mapping process

The previous work is improved in the next work by introducing Kolmogorov complexity. "The Google Similarity Distance" [23] deals with words and phrases acquire meaning from the way they are used in society, from their relative semantics to other words and phrases. A new theory of similarity between words and phrases based on information distance and Kolmogorov complexity is presented. The World Wide Web (WWW) is treated as the database, and Google as the search engine. The method is also applicable to other search engines and databases. This theory is then applied to construct a method to automatically extract similarity, the Google similarity distance, of words and phrases from the WWW using Google page counts. This model is improved by introducing a similarity ranking in literature[26].

"Scalable Probabilistic Similarity Ranking" is a scalable approach for probabilistic top-k similarity ranking on uncertain vector data. Each uncertain object is represented by a set of vector instances that is assumed to be mutually exclusive. The objective is to rank the uncertain data according to their distance to a reference object. The proposed framework computes instance and ranking position for each object, the probability of the object falling at that ranking position. The resulting rank probability distribution can serve as input for several state-of-the-art probabilistic ranking models. Existing approaches compute this probability distribution by applying the Poisson binomial recurrence technique of quadratic complexity. This complexity is reduced to a linear-time complexity with the same memory requirements in this framework. It is facilitated by incremental accessing of the uncertain vector instances in increasing order of their distance to the reference object.

3.4 Cluster Based Search

Search results on the Web are traditionally presented as a flat ranked list of documents. The main use for clustering is not to improve the actual ranking, but to give the user a quick overview of the results. Having divided the result set into clusters, the user can narrow down his search further by selecting a cluster. This resembles query refinement but avoids the need to query the search engine for each step. Evaluations done using the grouper system indicate that users tend to investigate more documents per query than in normal search engines. It is assumed that this is because the user clicks on the desired cluster rather than reformulating his query. The evaluation also indicates that once one interesting document has been found, users often find other interesting documents in the same cluster.

The majority of the current search engines generate a huge list in reply to a user query. This result is normally ranked by using ranking criteria such as page rank or relevancy to the query. However, this list is extremely inconvenient to users, since it expects them to look into each page sequentially in an exhaustive manner to find the relevant information. As a result, most users only search for an initial few Web pages on the list. Thus many other relevant information can be overlooked.

The clustering method [19] is one such solution to overcome this problem. Instead of a sequential list, it groups the search results into clusters and labels these with representative words for each cluster. These labeled clusters of search results are exposed to users. The clustering method provides benefits in terms of reduced size of information provided to the end users. The clusters of items with common semantic and/or other characteristics can guide users in refining their original queries, to zoom in on smaller clusters, and drill down through subgroups. Search result clustering has several specific requirements that may not be essential for other cluster algorithms.

First, search result clustering should allow fast clustering and rapid generation of a label on the fly, since it is an online process. This requirement can be met by adopting "snippets" rather than entire documents of a search result set. Second, labels annotated for clusters should be meaningful to users because they are presented to users as a general view of results. For this reason, recent search result clustering research focuses on selecting meaningful labels. This differs from general clustering which focuses on the similarity of documents.

The Lingo algorithm proposed uses frequent phrases to identify candidate cluster labels and then assigns snippets to these lables. The extension of this lingo algorithm by adding semantic recognition to the frequent extraction phase is presented in [1].

In this study, a collaborative proximity-based fuzzy clustering [27] is used to discover a structure of web information by a prudent reliance on the structures in the spaces of semantics and data. The method focuses on the reconciliation between the two separated facets of web information and a combination of results leading to a comprehensive data organization. The information arranged in this manner can provide an integral description of web resources. This style of processing is explicitly implied by the findings as to the relevance of the distinction made with regard to these two spaces. This approach dwells on some existing mechanisms of fuzzy clustering in particular fuzzy C-means to complete a thorough arrangement of collections of Semantic Web documents (SWDs)[14], according to their facet-based characteristics. Through the proposed collaborative clustering Fig.13, a collection of homogeneous clusters can be built. Given these constructs, to look at clusters of web resources which are useful to formulate the query and to drive a search toward some "similar" documents existing on the web.

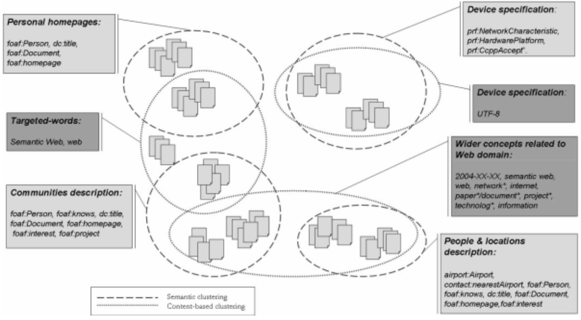
Personal homepages:
foaf:Person, dc:title, foaf:Document, foaf:homepage
Targeted-words:
Semantic Web, web
Communities description:
foaf:Person, foaf:knows, dc:title, foaf:Document, foaf:homepage, foaf:interest, foaf:project
Device specification:
prf:NetworkCharacteristic, prf:HardwarePlatform, prf:CcppAccept".
Device specification:
UTF-8
Wider concepts related to Web domain:
2004-XX-XX, semantic web, web, network*, internet, paper*/document*, project*, technolog*, information
People & locations description:
airport:Airport, contact:nearestAirport, foaf:Person, foaf:knows, dc:title, foaf:Document, foaf:homepage, foaf:interest
Semantic clustering
Content-based clustering

Fig. 13. Semantic and content-based clustering.

3.5 Crawler Based Search

"Swoogle" [14] is a crawler-based indexing and retrieval system for the Semantic Web. It extracts metadata for each discovered document, and computes relations between documents. Discovered documents are also indexed by an information retrieval system to find relevant documents and to compute the similarity among a set of documents. One of the interesting properties is computing *ontology rank*, a measure of the importance of a Semantic Web document. As shown in Fig.14, Swoogle's architecture can be broken into four major components: SWD discovery, metadata creation, data analysis, and interface. This architecture is data centric and extensible; components work independently and interact with one another through a database.

The ***SWD discovery*** component discovers potential SWDs throughout the Web and keeps up-to-date information about SWDs.

The ***metadata creation*** component caches a snapshot of a SWD and generates objective metadata about SWDs at both the syntax level and the semantic level.

The ***data analysis*** component uses the cached SWDs and the created metadata to derive analytical reports, such as classification of SWOs and SWDBs, rank of SWDs, and the IR index of SWDs.

The ***interface*** component focuses on providing data services to the Semantic Web community.

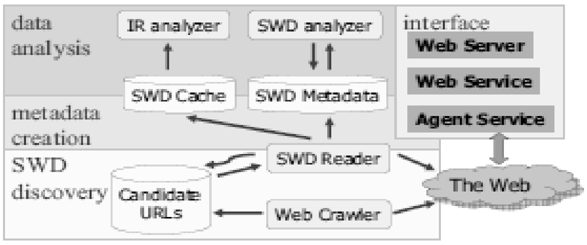

Fig 14.The architecture of Swoogle

Swoogle is improved by adding user preferences and interests to provide user a set of personalized results. In this paper the author proposes, architecture for a Personalized Semantic Search Engine (PSSE) [21]. PSSE is a crawler-based search engine that makes use of multi-crawlers to collect resources from both semantic as well as traditional web resources. In order to reduce processing time, web pages' graph is clustered, then clusters are annotated using document annotation agents that work in parallel. Annotation agents use methods of ontology matching to find resources of the semantic web as well as information extraction techniques. System ranks resources based on a final score that's calculated based on traditional link analysis, content analysis and a weighted user profile for more personalized results.

PSSE Architecture: As Fig.15 depicts, the processes of PSSE are separated into an offline and an online part. The offline part includes crawling and preprocessing processes. The online phase includes query processing and result ranking.

Offline Phase In this phase, crawling the World Wide Web and preprocessing of crawled pages take place.

Crawler PSSE uses Multi-crawlers (web spiders) that traverse World Wide Web, collect web resources and store them in database. Crawlers work with the aid of information extraction techniques to find link information in the retrieved pages.

Preprocessor The preprocessor is used to maintain resources that are downloaded from Web sites. The main task of query Indexer and link analyzer is to cluster the crawled web documents to enable parallel processing. This can be done in three steps: first indexer and link analyzer builds a graph of the crawled pages. Link analysis is then performed to calculate authoritativeness of web pages. And finally the graph is clustered by identifying its connected components. These clusters are then annotated by annotation agents that work in parallel to reduce processing time. Afterwards, annotations are weighted so as to determine their relevancy to web resource using term relevancy evaluator.

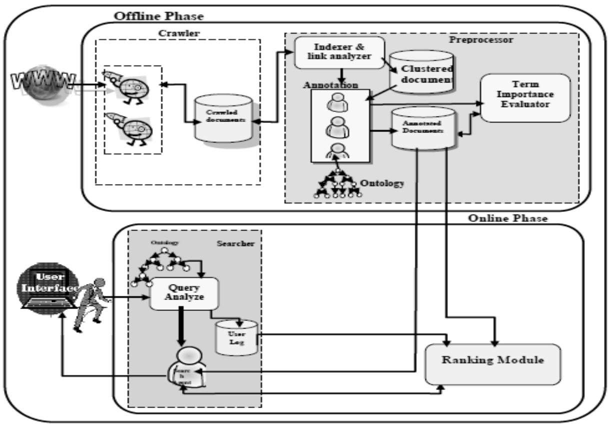

Fig 15. Architecture of PSSE

3.6 Other Directions

An approach based on metadata is discussed by Fabio Silva et al [8]. This work proposes a model to find information items with similar semantic content that a given user's query. The information items internal representation is based on user interest groups, called "semantic cases". The model also defines a similarity measure for ordering the results based on semantic distance between semantic cases items.

An annotation process extracts the metadata which is used to build the internal representation of documents and queries. Finally the matching process that uses concepts is used to find related documents and a semantic similarity function for the retrieval results ranking (Fig. 16). The main limitation of this model is the incompleteness of the conceptualization. The annotation process must be supported by ontology learning, to discover new items.

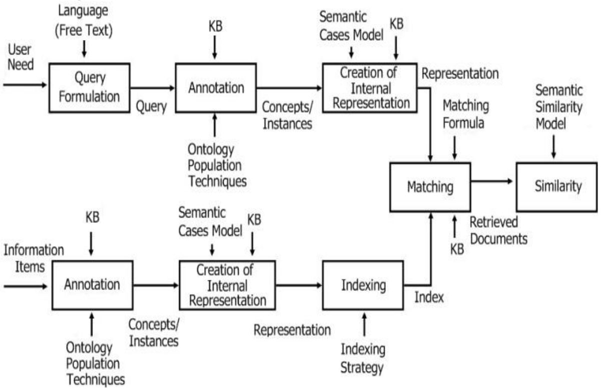

Fig 16. Overview of the proposed information retrieval process

The architecture for Developing a semantic-enable information retrieval mechanism [16] handles the processing, recognition, extraction, extensions and matching of content semantics to achieve the following objectives. i. Analyse and determine the semantic features of content, to develop a semantic pattern that represents semantic features of the content, and to structuralize and materialize semantic features; ii. Analyse user's query and extend its implied semantics through semantic extension so as to identify more semantic features for matching iii) Generate contents with approximate semantics by matching against the extended query to provide correct contents to the queriest.

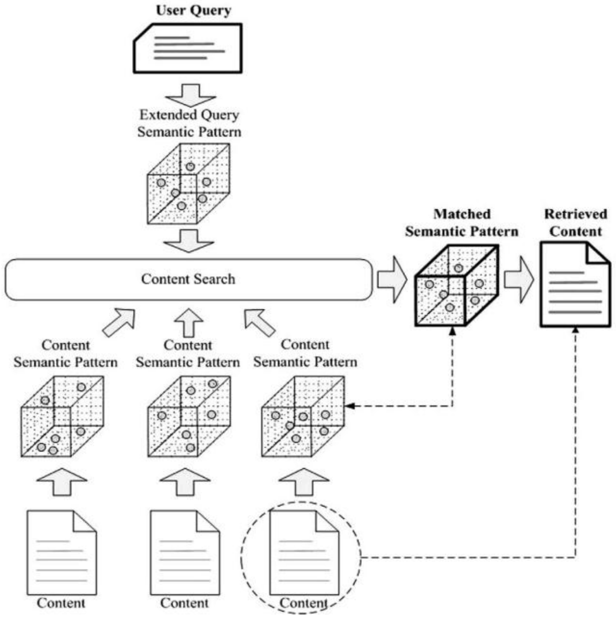

Fig. 17. Scenario of semantic-enable information retrieval

This architecture contains the core technologies such as Semantic determination and extraction, Semantic extension, Semantic pattern clustering and matching. In addition to semantic-based information retrieval, the proposed system has two main features: i) Latent semantic analysis to generate more semantics for matching, thereby solving the problem of insufficient information for query; ii) Semantic clustering model which identifies the corresponding document category for the query and then performs content matching in that category thereby improving matching accuracy.

An overview of Semantic Search Systems [29] discussed with the help of a framework which has six components responsible for data acquisition, knowledge acquisition, data integration and consolidations, semantic search mechanisms, semantic search services , and result presentation.

The ***Semantic data acquisition*** will provide different solutions to collect all the structured, semi structured and unstructured data. The collected data is transformed into structured data using ***Knowledge acquisition*** component.

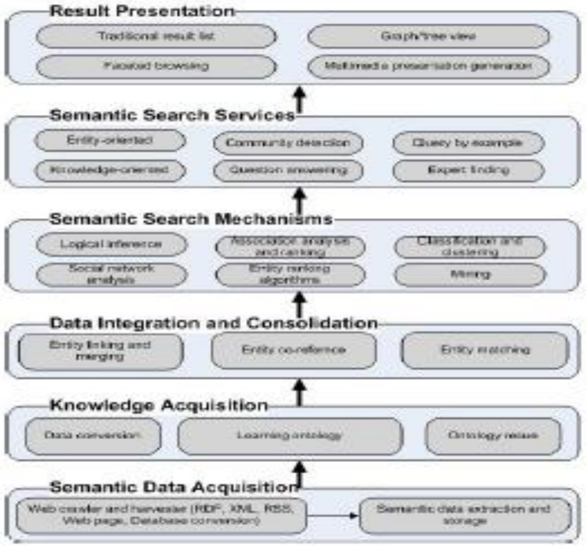

Fig 18. A Semantic Search Framework

The ***Data integration and consolidation*** component summarises solutions for a problem arisen from the previous stage. ***Search mechanisms*** component deals with various techniques based on which semantic search services are implemented. ***Semantic Search Services*** provide an abstract model of the functionalities a semantic search engine offers. Finally the ***Result Presentation*** component presents the search results to the user.

4. Conclusion

In this paper, we survey the various architectures of Semantic Search Systems and classify them in five dimensions: Relation Based, Ontology Based, Similarity Based, Cluster Based, Crawler Based Search systems. The first dimension process the user’s query based on the keyword- concept pair. The second dimension will find the relevant Ontology to the user. The third criteria ranks the Ontologies based on the rank calculated and arranged, the most relevant ontology is submitted to the user. In the fourth criteria, instead of linear list the results are presented in the form of clusters with appropriate labels. The last dimension makes use of the crawler to collect the semantic documents and to find the relevant information on the retrieved paper.

There are several points to make from this survey as a future direction. First the semantic searching mainly focused on the trust and the quality of knowledge which varies largely from source to source. Effective ranking algorithms are needed to distill most trustworthy and quality information. The second aspect is ontology-based research focused mainly on integrity of the Domain Ontology, Automatic Ontology Evolution and Ontology Learning. Since the web is decentralized and heterogeneous, even on the same domain it seems impossible for all web pages to use the same ontology. So study on semantic interoperability will be needed. The third aspect is assigning weights relies on the user explicitly assigning numerical weights to properties through the query interface and hence imposes some overhead to the users. Methods should be explored to assign weights automatically through relevance feedback strategy and predicting users preference. Another promising direction is to incorporate rules to support more powerful reasoning based intelligent semantic search.

References

[1] Ahmed Samesh, Amar Kadray ,“Semantic Web Search Results Clustering Using Lingo and WordNet” , International Journal of Research and Reviews in Computer Science (IJRRCS) Vol. 1,No.2, June 2010.

[2] Aleman-Meza B, Halaschek C, Arpinar IB, Sheth “A Context-aware semantic association ranking”, Semantic web and databases workshop proceedings. Berlin, Germany, September pp.7–8, 2003.

[3] Amit Pisharody, Howard E.Michel,“A Search Engine Technique Using Relation Based Keywords”, Proceedings of the 2005 International Conference on Artificial Intelligence, IC-AI 05.

[4] Baeza-Yates R A and B A Riberiro-Neto, "Modern Information Retrieval",ACM Press/Addision-Wesley,1999.

[5] Broekstra J, EhrigM, Haase P, van Harmelen F,KampmanA, Sabou M,et al. "A metadata model for semantics-based peer-to-peer systems", Proceedings of the WWW'03 workshop on semantics in peer-to-peer and grid computing. Budapest, Hungary, pp.20–24, May 2003.

[6] Burners-Lee T, J Hendler and O Lassila, " The Semantic Web", Scientific American, Vol.284,No.4.2001.

[7] Darijus Strasunskas and Stein Tomassen "On Variety of Semantic Search Systems and Their Evaluation Methods" International Conference on nformation Management and Evaluation (ICIME 2010), pp. 380-387, Mar 2010.

[8] Fabio Silva, Rosario Girardi, and Lucas Drumond "An Information Retrieval Model for the Semantic Web" 2009 Sixth International Conference on Information Technology: New Generations,2009.

[9] Fabrizio Lamberti, Andrea Sanna, Claudio Demartini, " A Relation-Based Page Rank Algorithm for Semantic Web Search Engines " IEEE Trans. Knowledge and Data Eng., vol. 21, No.1, pp. 123-135, Jan. 2009.

[10] Gopinath Ganapathy1 and S. Sagayaraj "Studies on Architectural Aspects of Searching using Semantic Technologies" International Journal of Research and Reviews in Computer Science (IJRRCS) Vol. 1, No. 2,pp. 119-126, June 2010.

[11] Guha.R, McCool.R and Miller.E, "Semantic Search", International Conference on World Wide Web, pp.700-709,2003.

[12] Guobing Zou Bofeng Zhang Yanglan Gan Jianwen Zhang "An Ontology-based Methodology for Semantic Expansion Search", Fifth International Conference on Fuzzy Systems and Knowledge Discovery 2008.

[13] Lee.J, Kim.M, and Lee.Y, "Information Retrieval Based on Conceptual Distance in IS-A Hierarchies," Documentation, vol 49. pp.188-207,1993.

[14] Li Ding, , Finin.L, , Joshi.T.W, , Pan.A, Scott Cost.R, Peng.R, Reddivari.Y Doshi.P, Sachs.S "Swoogle: a search and metadata engine for the semantic web", CIKM 2004, pp.652-659,2004.

[15] Lixin Hana,b,c, Linping Suna, Guihai Chenb, Li Xieb "ADSS: An approach to determining semantic similarity Advances in Engineering Software" Elsevier, Science Direct 37, pp. 129–132,(2006).

[16] Ming-Yen Chen, Hui-Chuan Chu, Yuh-Min Chen "Developing a semantic-enable information retrieval mechanism" Elsevier, Expert Systems with Applications 37 pp. 322–340,(2010).

[17] Pablo Castells, Miriam Ferna´ndez, and David Vallet " An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval" IEEE Transactions On Knowledge And Data Engineering, Vol. 19, No. 2, pp. 261-272, February 2007.

[18] Pekar V, Staab S. "Word classification based on combined measures of distributional and semantic similarity", Proceedings of the research note sessions of the 10th conference of the European chapter of the association for computational linguistics (EACL'03). Budapest,Hungary, pp. 12–17 ,April 2003.

[19] Ramesh Singh, Dhruv Dhingra, and Aman Arora "SSCHISM—A Web search engine using semantic taxonomy "IEEE Potentials, pp 36-40,2010.

[20] Ravi Sankar V, A.Damodaram and P.Radha Krishna "Ranking Ontologies Based on OWL Language Constructs", Information Technology Journal 9(3):,ISSN 1812-5638 pp. 553-560,2010.

[21] Riad A.M., Hamdy.K Elminir., Mohamed Abu ElSoud, Sahar. F. Sabbeh. "PSSE: An Architecture For A Personalized Semantic Search Engine" ,International Journal on Advances in Information Sciences and Service Sciences Volume 2,Number 1, pp. 102-112 ,March 2010.

[22] Rodriguez M, Egenhofer M. "Determining semantic similarity among entity classes from different ontologies", IEEE Transactions on Knowledge and Data Engineering 15(2), pp. 442–56,2003.

[23] Rudi L. Cilibrasi and Paul M.B. Vita´ nyi "The Google Similarity Distance" IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 3, pp.370-383, March 2007.

[24] Song Jun-feng, Zhang Wei-ming, Xiao Wei-dong, Li Guo-hui, Xu Zhen-ning "Ontology-Based Information Retrieval Model for the Semantic Web "International Symposium on Intelligent Information Technology Application Workshops 2008.

[25] Sridevi.U.K, Nagaveni "Ontology based Similarity Measure in Document Ranking", International Journal of Computer Applications (0975 - 8887) Volume 1 – No. 26 pp.135-139,2010.

[26] Thomas Bernecker, Hans-Peter Kriegel, Nikos Mamoulis, Matthias Renz, and Andreas Zuefle "Scalable Probabilistic Similarity Ranking in Uncertain Databases" IEEE Transactions on Knowledge and Data Engineering, Vol. 22, No. 9, pp.1234-1246, September 2010.

[27] Vincenzo Loia, Witold Pedrycz, and Sabrina Senatore"Semantic Web Content Analysis: A Study in Proximity-Based Collaborative Clustering" IEEE Transactions on Fuzzy Systems, Vol.15, No. 6, pp.1294-1312 ,December 2007.

[28] Wei-Dong Fang, Ling Zhang, Yan Xuan Wang, Shou-Bin Dong " Toward a Semantic Seach Engine Based On Ontologies" Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, pp.18-21, August 2005.

[29] Wei Wang, Payam M. Barnaghi, Andrzej Bargiela, "Search with Meanings:An Overview of Semantic Search Systems", International journal of Communications of SIWN, Vol. 3, pp. 76-82, June 2008.

[30] Xiaomeng Su , Jon Atle Gulla "An information retrieval approach to ontology mapping Data & Knowledge Engineering" Elsevier, ScienceDirect 58 pp. 47–69 ,(2006) .

[31] Yi Jin, Zhuying Lin, Hongwei Lin" The Research of Search Engine Based on Semantic Web " International Symposium on Intelligent Information Technology Application Workshops,2008.

[32] Yi Zhang, Wamberto Vasconcelos, Derek Sleeman "OntoSearch:An Ontology Search Engine " Research and Development in Intelligent Systems XXI 2005, Session 1a, DOI: 10.1007/1-84628-102-4_5,pp. 58-69,2005.

[33] Yuangui Lei, Victoria Uren, Enrico Motta, "SemSearch: A Search Engine for the Semantic Web", EKAW 2006, Springer, pp:238-245, 2006.

[34] Yufei Li, Yuan Wang, and Xiaotao Huang "A Relation - Based Search Engine in Semantic Web," IEEE Trans. Knowledge and Data Eng., vol. 19, No.2,pp. 273-282, Feb. 2007.

[35] Zheng Gu, Song-Nian Yu, "Ontology-Based Inverted Tables in Information Retrieval System", Third International conference on Semantics,Knowledge and Grid 2007.

[36] http://www.w3.org/RDF/, Resource Description Framework (RDF); World Wide Web Consortium; August, 2003.

Author Biographies

Muthuram Thangaraj received his post-graduate degree in computer science from Alagappa University, Karaikudi, M.Tech. degree in Computer Science from Pondicherry University and Ph.D. degree in Computer Science from Madurai Kamaraj University, Madurai,TN,South India in 2006. He is now the Associate Professor of Computer Science Department at M.K.University. He is an active researcher in Webmining, Semantic Web and Inforamtion Retrieval and has published more than 35 papers in Journals and Conference Proceedings.

Sujatha G was born on 14.7.71 in Madurai. She received her M.C.A. degree in 1994 from Alagappa University, Karaikudi,M.Phil. in Computer Science from Mother Teresa University, Kodaikanal in 2000.Currently she is an Assistant Professor in Computer Science, Sri Meenakshi Govt. College for Women, Madurai, Tamil Nadu, India. Her research interests are focused on Semantic Search and the Information Retrieval. sujisekar05@rediffmail.com.