Web Mining and Analysis on Semantic Web Search Engine

B.SANTHOSH KUMAR
Department of Computer Science and Engineering
C S I College of Engineering,
Ketti, The Nilgiris-643215, Tamilnadu, India
b.santhoshkumar@gmail.com

M.MOHAN
Department of Information Technology
C S I College of Engineering,
Ketti, The Nilgiris-643215, Tamilnadu, India
m.mohansan@gmail.com

Abstract— **The Web search engines plays a critical role in the mining of data from the large number of web information's in the form of web pages. The existing Semantic web search engines are failed to retrieve the web pages with the desired amount of accuracy. The ranking needs to work on whole of the annotated knowledge database. The proposed system uses the layered architecture which will increase the information retrieval accuracy using relations. But in this relation-based page rank algorithm to be used in conjunction with Semantic web search engine. It emphasize on the information extracted from the user queries on annotated resources. Relevance between queries is measured in terms of probability that a retrieved resource actually contains the relations based on the user query. It tends to produce results in terms of both time complexity and accuracy.**

Keywords- **Semantic Web, Knowledge Retrieval, Search Process**

I. INTRODUCTION

The search engines are comes to play ever a more critical role because of the tremendous growth of information available to end users through the Web. It is always less uncommon that obtained result sets provide a burden of useless pages. The next-generation Web architecture, represented by the Semantic Web, provides the layered architecture possibly allowing overcoming this limitation. Semantic search engines have been proposed, which allow increasing information retrieval accuracy by exploiting a key content of Semantic Web resources that is relations. In order to rank results, most of the existing solutions need to work on the whole annotated knowledge base.

The aim of this project is to show how to make use of relations in Semantic Web page annotations with the aim of generating an ordered result set, where pages that best fit the user query are displayed first. To evaluate the feasibility of the proposed approach, first constructed a controlled Semantic Web environment. To do, selected the well-known *travel.owl* ontology written in the OWL language and modified it by adding new relations in order to make it more suitable for demonstrating system functionality. We then created a knowledge base by either downloading or automatically generating a set of web pages in the field of tourism, and embedded into RDF semantic annotations based on the *travel.owl* ontology.

Finally, designed the remaining modules of the architecture, including a Webpage database, a crawler application, a knowledge database, an OWL parser, a query interface and the true search engine module embedding the proposed into ranking logic. The crawler application collects annotated Web pages from the Semantic Web (in this case, represented by the controlled environment and its Web page collection) including RDF metadata and originating OWL ontology. RDF metadata are interpreted by the OWL parser 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 details of the system workflow will be provided in the following sections, starting with the query definition process, since it was through the analysis of its dynamics that came to the identification of our ranking strategy. The Query

Definition Process, in a traditional search engine like Google, a query is specified by giving a set of keywords, possibly linked through logic operators and enriched with additional constraints (i.e. document type, language, etc). Semantic search engines are capable of exploiting concepts (and relations) hidden behind each keyword together with natural language interpretation techniques to further refine the result set.

The user specifies a query by entering a keyword and selecting a concept from a pull-down menu containing ontology classes of the *travel.owl* ontology organized in a hierarchical fashion. It is worth observing that the current implementation is not able to handle multiple ontology's describing the same domain. The search logic, would require the integration of one of the existing techniques for mapping or merging/translating the heterogeneous ontology's, which would result in the definition of a set of mapping rules or in the creation of a novel (possibly extended) ontology, respectively.

The user interaction, having an extended ontology would increase the need for a preprocessing step enabling automatic identification of keyword-concept pairs. The page contains exactly those relations that are of interest to the user, and as a consequence, that the page is actually the most relevant with respect to user query. The idea is to define a "ranking criterion" based on an estimate of the probability that keywords/concepts within an annotated page are linked one to the other in a way that is the same (or at least that is similar) 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 sub graph), and of each annotated page containing queried concepts/keywords (both in terms of annotation graph and page sub graph).

II. RELATED WORKS IN SEMANTIC WEB SEARCH

The aim of this paper is to show how to make use of relations in Semantic Web page annotations with the aim of generating an ordered result set, where pages that best fit the user query are displayed first. The idea of exploiting ontology based annotations for information retrieval is not new [5], [6], [7], [14]. The first works did not focus on semantic relations, which are considered (and expected) to play a key role in the Semantic Web [9], [13]. It has been recently outlined that in order to fully benefit on semantic contents, a way for achieving relation based ranking has to be found [3], [9], [11], [15]. One of the first attempts to enhance Semantic Web search engines with ranking capabilities is reported in [11].

To define a similarity score measuring, the distance between the systematic descriptions of both query and retrieved resources. They first explode an initial set of relations (properties) by adding hidden relations, which can be inferred from the query. In the ontology and annotation graphs, concepts and relations are translated into graph nodes and edges, respectively. To do, the notions of query sub graph and page sub graph have to be introduced. In a query sub graph, nodes are represented by concepts that have been specified within the query. Nodes/concepts are linked by an (weighted) edge only, if there exists at least one relation between those concepts in the ontology.

The weight is represented by the actual number of relations. A page sub graph is built based on the annotation associated to the page itself. It is predictable that the number of relations will largely exceed the number of concepts [2], its applicability in real contexts is severely compromised. A similar approach, aimed at measuring the relevance of a semantic association (that is, a path traversing several concepts linked by semantic relations) is illustrated in [15]. Semantic approach suffers from the same limitations of [11], queries have to be specified by entering both concepts and relations, and ambiguity is measured over each relation instance. The idea of exploring the set of relations that are implicit in the user's mind has been pursued in many works.

In [10], ontology-based lexical relations like synonyms, antonyms, and homonyms between keywords have been used to "expand" query results. Search is targeted to the Web, rather than to the Semantic Web. In [16], a similar approach has been integrated into artificial intelligence methodologies to address the problem of query answering. In [4], query logs are used to construct a user profile to be later used to improve the accuracy of Web search. Semantic Web search from the point of view of the user's intent has been addressed also in [8] and [17], where the authors present two methodologies for capturing the user's information need by trying to formalize its mental model.

They analyze keywords provided during query definition, automatically associate related concepts, and exploit the semantic knowledge base to automatically formulate formal queries. A slightly different methodology has been exploited in Sem Rank [3]. The basic idea is still to rank results based on how predictable a result might be for the user but based on how much information is conveyed by a result, thereby giving a sense of how much information a user would gain by being informed about the existence of the result itself. Candidate relation-keyword set (CRKS) to be submitted to the annotated database, which can significantly reduce the presence of uninteresting pages in the result set.

It is worth observing that the strategy behind Onto Look only allows us to empirically identify relations among concepts that should be less relevant with respect to the user query. This information is used to reformulate the user query by including only a subset of all the possible relations among concepts, which is later used to retrieve web pages from the annotated database. The user is not requested to specify relations of interest during query definition. The effectiveness of the approach is strongly limited by the fact that there does not exist any ranking strategy. Many other statistical and text-matching techniques are used together with Page Rank. Page Rank can be used in conjunction with [9] to exploit relevance feedback and post process the result set. But the use of the remaining techniques is not feasible since they cannot be reasonably applied into a concept-relation-based framework where ontology is predominant on pure text.

The search engine logic should only need to know the structure of the underlying ontology and of the Web page to be ranked in order to compute the corresponding relevance score. The effective performance can be achieved in heterogeneous real frameworks. It is worth observing that the proposed approach could be easily seen as an extension of [9]. It does not represent an alternative to any of the approaches above, but rather, they can be regarded as complementary to our solution. The availability of an ad hoc language allowing the user to pre process the graph and reduce the region of interest [15] could be integrated in our approach as a pre processing step. Similarly, the availability of instruments for inferring concepts of interest starting from a pure keyword based query [12] can be helpful to limit the amount of knowledge of the underlying ontology requested to the user. Finally, the proposed technique is not intended to replace the ranking strategies of actual search engines. It should be understood as a pre-processing step to produce a semantic aware ordered result set to be later (or simultaneously) treated with existing (popular) techniques in order to come to an increased hit ratio in user query processing.

III.SEMANTIC SEARCH ON THE WEB

This project Semantic Web search engine is based on semantic search. Search engine is constitute the most helpful tools for organizing information and extracting knowledge from the Web and it is not uncommon that even the most renowned search engines return result sets including many pages that are definitely useless for the user. The relations among concepts embedded into semantic annotations can be effectively exploited to define a *ranking strategy* for Semantic Web search engines. Semantic Web approach only relies on the knowledge of the user query, and the Web page to be ranked, and the underlying ontology. It allows effectively manage the search space and to reduce the complexity associated with the ranking task. It is based on ontology lexical relations like synonyms, antonyms, and homonyms between keyword (but not concept) have been used to “expand” query results.

The search is targeted to the Web, rather than to the Semantic Web. The existing page ranking algorithms can be used to order the obtained result set, it is worth remarking that this is not completely true. In fact, a ranking strategy likes the Page Rank used by Google, is only one of the ranking algorithms used to organize results to be displayed to the user. The crawler application collects annotated Web pages from the Semantic Web (in this case, represented by the controlled environment and its Web page collection) including RDF metadata and originating OWL ontology. In Ranking Methodology, the search engine logic accesses the Web page database, constructs the initial result set including all those pages that contain queried keywords and concepts, and computes the query sub graph. For each page in the result set, the page sub graph is computed. Starting from each sub graph, all page spanning forests are generated and used to compute the page score. Web pages are associated to relevance classes, and the final (ordered) result set is constructed.

A. Web Crawler

First deals with the creation of effective Web crawler, a Web crawler (also known as a Web spider or Web robot) is a program or automated script which browses the World Wide Web in a methodical automated manner. Web crawler starts with a list of URLs to visit, called the seeds. As the crawler visits these URLs, it identifies all the hyperlinks in the page and adds them to the list of URLs to visit, called the crawl frontier. URLs from the frontier are recursively visited according to a set of policies. In this Search Engine, the WebCrawler will start with some seeds and it will select the pages using some filters and policies. To create a simple Search Engine the crawler will be programmed to download given index page related pages only.

The behavior of a Web crawler is the outcome of a combination of policies:

- Selection policy that states which pages to download.

- Re-visit policy that states when to check for changes to the pages.
- Politeness policy that states how to avoid overloading websites.
- Parallelization policy that states how to coordinate distributed web crawlers.

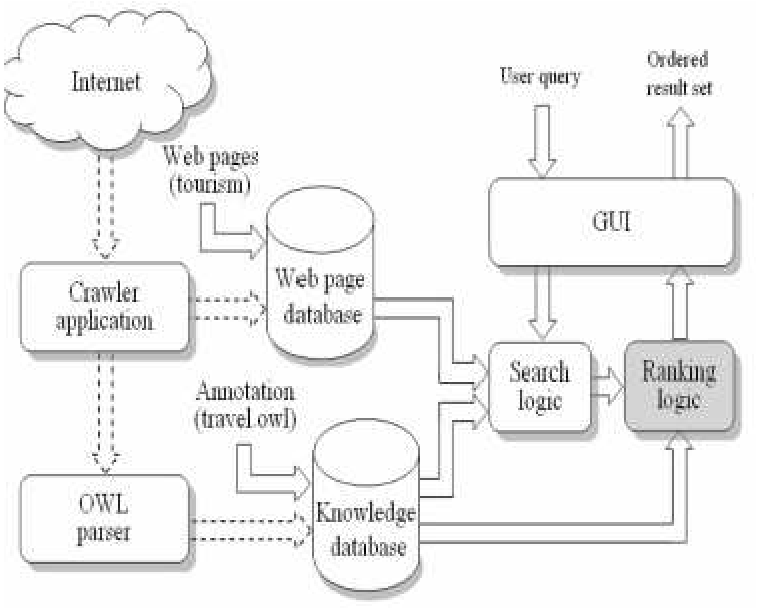

Fig.1 Semantic Web Infrastructure (Prototype architecture)

The crawler application collects annotated Web pages from the Semantic Web (in this case, represented by the controlled environment and its Web page collection) including RDF metadata and originating OWL ontology. RDF metadata are interpreted by the OWL parser 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 details of the system workflow will be provided in the following sections, starting with the query definition process, since it was through the analysis of its dynamics that we came to the identification of our ranking strategy. Spiders are visits a web page, read it and then follow links to other pages within the site. Everything, the spider finds, and then goes into the second part of the search engine the index and it containing a copy of every web page. If a web page changes then this book is updated with new information.

Search Engine Software is the program that sifts through the millions of pages, recorded in the index to find matches to a search and rank them in order of what it believes is most relevant .Search for anything using your favourite crawler based search engine. Instantly the search engine will sort through the millions of pages it knows about and present with ones that match the topic. The matches will even be ranked, so that the most relevant one comes first.

B. Indexer

Search engine will crawl all the domains and sub domains added to its' URL list. It processes each of the pages as it crawls in order to compile a massive index of all the words it sees and their location on each page. It may take 100 or even 1000 jumps for the crawler to find a page, but if the page is linked from another page it will be indexed.

Search engine indexing collects, parses, and stores data to facilitate fast and accurate information retrieval. Index design incorporates interdisciplinary concepts from linguistics, cognitive psychology, mathematics, informatics, physics and computer science. An alternate name for the process in the context of search engines designed to find web pages on the Internet is Web indexing.

Policies followed by indexer,

- The search engine honors **robots.txt** files, **META robot** entries.

- Web server may have blocked either the directory or Web server that contains the page you are looking for.
- We have allowed only certain file formats from be indexed in order to maintain a high level of integrity and quality user experience with search results.

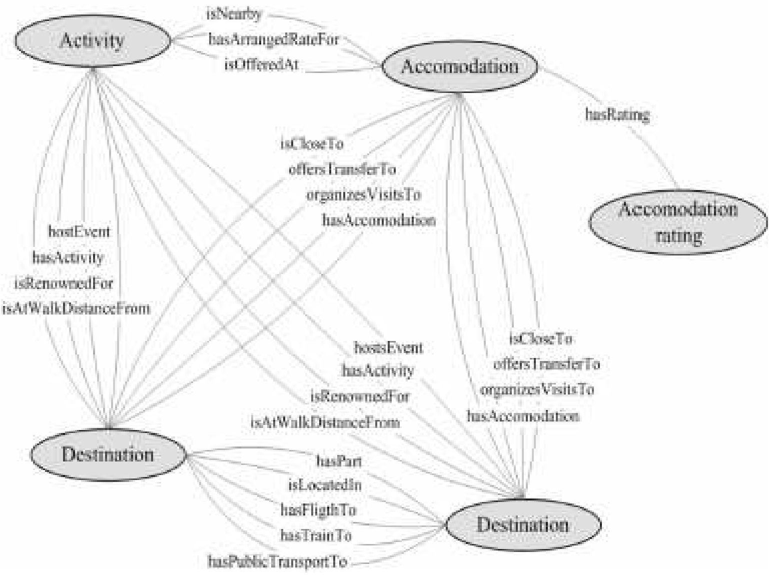

Fig.2 The Graph Based Representation For Ontology

C. Interface Management Module

1) Search Manager

The Search Manager induces accurate search results by bringing the related domain semantic information from the ontology server based on the query input by the user and requesting the user for a second query. When a multiple number of domain semantic information has been found the manager suggests the subject words and descriptions of the semantic information for the user to select.

2) Classification Manager

The Classification Manager's method of searching by subject is quite distinctive compared to the current layered structure method. Searching by subject determines the relations between terms based on the RDF documents and enables more precise and efficient search for documents by applying a flexible network structure.

3) User Interface Manager

The User Interface Manager provides various users' search input screens, ontology information selecting screens and final information search results screens.

D. Engine Ontology Server

E-engine Ontology Server (world map) is placed above the syntactic layer (XML) and semantic layer (RDF). It is a systematic method of expression that can improve the present condition where information is processed simply as data and the semantic context must be provided by man and allow information to have value as knowledge.

E. Page Ranking

Page Ranking is the main module in which the crucial page ranking is applied to the web page results that are filtered by above modules. In order to rank the popularity of Semantic Web documents, adopt the surfing model in which a rational surfer always recursively pursues the definition of classes and properties for complete understanding of a given RDF graph. We propose a relation based page rank algorithm to search algorithm to make the ranking effective. The ontology defined for a domain, a graph based representation can be designed where OWL classes are mapped into graph vertices and OWL relation properties are mapped into graph edges. Thus, the existing relations between couples of concepts in the domain are depicted by means of connected vertices in the graph. We call it the ontology graph G. Search engines for both the conventional Web and the Semantic Web involve the same set of high-level tasks are discovering and harvesting documents, processing search queries from users and agents, ranking search results, caching and archiving documents, and providing human interfaces and software. A formal model for the proposed ranking strategy will be provided, by taking into account all the critical situations that could be envisioned. The ranking will be applied to the result produced by the given query and the top ranked lists are shown as a result.

F. Semantic Management Module

The Semantic Management Module, the information extracting agent is used for extracting related Web pages, and then the wrapper is used to return XML documents based on the material. The Automatic Classification Module is used to automatically classify the pages and then the results are stored in the Content DB Server.

IV. CONCLUSION AND FUTURE WORK

Web architecture represented by the Semantic Web will provide adequate instruments for improving search strategies and enhance the probability of seeing the user query satisfied without requiring tiresome manual refinement. Actual methods for ranking the returned result set will have to be adjusted to fully exploit additional contents characterized by semantic annotations including ontology-based concepts and relations. Several ranking algorithms for the Semantic Web exploiting relation-based metadata have been proposed. They mainly use page relevance criteria based on information that has to be derived from the whole knowledge base, making their application often unfeasible in huge semantic environments.

A novel ranking strategy that is capable of providing a relevance score for a Web page into an annotated result set by simply considering the user query, the page annotation, and the underlying ontology. Page relevance is measured through a probability-aware approach that relies on several graph-based representations of the involved entities. By neglecting the contribution of the remaining annotated resources, a reduction in the cost of the query answering phase could be expected. Despite the promising results in terms of both time complexity and accuracy, further efforts will be requested to foster scalability into future Semantic Web repositories based on multiple ontology, characterized by billions of pages, and possibly altered through next generation “semantic” spam techniques.

ACKNOWLEDGMENT

This research is supported by the management of ***C.S.I College of Engineering, Ketti***, The Nilgiris, whose support we are pleased to acknowledge. We are also grateful to our colleagues in CSICE for useful discussions, and thank to our beloved CSICE students.

REFERENCES

[1] 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, Jan. 2009.

[2] B. Aleman-Meza, C. Halaschek, I. Arpinar, and A. Sheth, “A Context- Aware Semantic Association Ranking,” *Proc. First Int’l Workshop Semantic Web and Databases* (SWDB ’03), pp. 33-50, 2003.

[3] K. Anyanwu, A. Maduko, and A. Sheth, “SemRank: Ranking Complex Relation Search Results on the Semantic Web,” *Proc. 14th Int’l Conf. World Wide Web* (WWW ’05), pp. 117-127, 2005.

[4] R. Baeza-Yates, L. Caldero´n-Benavides, and C. Gonza´lez-Caro, “The Intention behind Web Queries,” *Proc. 13th Int’l Conf. String Processing and Information Retrieval (SPIRE ’06),* pp. 98-109, 2006.

[5] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv, “XSEarch: A Semantic Search Engine for XML,” *Proc. 29th Int’l Conf. Very Large Data Bases*, pp. 45-56, 2003.

[6] L. Ding, T. Finin, A. Joshi, R. Pan, R.S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs, “Swoogle: A Search and Metadata Engine for the Semantic Web,” *Proc. 13th ACM Int’l Conf. Information and Knowledge Management* (CIKM ’04), pp. 652-659, 2004.

[7] R. Guha, R. McCool, and E. Miller, “Semantic Search,” *Proc. 12th Int’l Conf. World Wide Web* (WWW ’03), pp. 700-709, 2003.

[8] Y. Lei, V. Uren, and E. Motta, “SemSearch: A Search Engine for the Semantic Web,” *Proc. 15th Int’l Conf. Managing Knowledge in a World of Networks* (EKAW ’06), pp. 238-245, 2006.

[9] Y. Li, Y. Wang, and X. Huang, “A Relation-Based Search Engine in Semantic Web,” *IEEE Trans. Knowledge and Data Eng.*, vol. 19, no. 2, pp. 273-282, Feb. 2007.

[10] A. Pisharody and H.E. Michel, “Search Engine Technique Using Keyword Relations,” *Proc. Int’l Conf. Artificial Intelligence (ICAI ’05), pp. 300-306, 2005.*

[11] T. Priebe, C. Schlager, and G. Pernul, “A Search Engine for RDF Metadata,” *Proc. 15th Int’l Workshop Database and Expert Systems Applications* (DEXA ’04), pp. 168-172, 2004.

[12] C. Rocha, D.Schwabe, and M.P. Aragao, “A Hybrid Approach for Searching in the Semantic Web,” *Proc. 13th Int’l Conf. World Wide Web* (WWW ’04), pp. 374-383, 2004.

[13] A. Sheth, B. Aleman-Meza, I.B. Arpinar, C. Bertram, Y. Warke, C. Ramakrishanan, C. Halaschek, K. Anyanwu, D. Avant, F.S. Arpinar, and K. Kochut, “Semantic Association Identification and Knowledge Discovery for National Security Applications,” J. *Database Management*, vol. 16, no. 1, pp. 33-53, 2005.

[14] N. Stojanovic,“An Explanation-Based Ranking Approach for Ontology- Based Querying,” *Proc. 14th Int’l Workshop Database and Expert Systems Applications,* pp. 167-175, 2003.

[15] N. Stojanovic,R. Studer, and L. Stojanovic, “An Approach for the Ranking of Query Results in the Semantic Web*,” Proc. Second Int’l Semantic Web Conf*. (ISWC ’03), pp. 500-516, 2003.

[16] R. Sun, H. Cui, K. Li, M.Y. Kan, and T.S. Chua, “Dependency Relation Matching for Answer Selection*,” Proc. ACM SIGIR* ’05, pp. 651-652, 2005.

[17] T. Tran,P. Cimiano, S. Rudolph, and R. Studer, “Ontology-Based Interpretation of Keywords for Semantic Search,” *Proc. Sixth Int’l Semantic Web Conf*., pp. 523-536, 2007.