International Journal of Computer and Internet Security.
ISSN 0974-2247 Volume 3, Number 1 (2011), pp. 25-33

http://www.irphouse.com

A Keyword-Based on Semantic Web Search Engine

[1]P. Indira Priya and [2]D.K. Ghosh

[1,2]Department of Computer Science and Engineering
[1]Tagore Engineering College, Chennai, India
[2]V.S.B. Engineering College, Karur, India
E-mail: [1]pindirapriya@gmail.com, [2]drdaliml@yahoo.co.in

Abstract

Web search engines have become one of the most helpful tools for obtaining useful information from the Internet. The most popular search engines cannot produce satisfactory Web search results. Web search engine is a crawler-based indexing and retrieval system for the semantic Web, i.e., for Web documents in RDF or OWL. It extracts metadata for each discovered document, and computes relations between documents. The documents are also indexed by an information retrieval system which can use keywords to find relevant documents and to compute the similarity among a set of documents. One of the properties computed for each semantic Web document is its rank, a measure of the documents importance on the semantic Web. This paper proposes both architecture and a prototype of a keyword based on Semantic Web Search Engine.

Keywords: Semantic Web, Web Search.

Introduction

The World Wide Web (WWW) represents one of the largest, distributed, heterogeneous, semi structured, repositories of information. The users are experiencing increased frustration with searching the Web because of the difficulty of finding useful, relevant information from the huge volume of data deposited in the Web, reducing the effectiveness of the Web. The Semantic Web, currently in the form of a Web of Semantic Web documents (i.e. online documents written in RDF and OWL), is essentially a web universe parallel to the web of online documents. Semantic Web document (SWD) is well known for its semantic annotation and meaningful reference. A search engine customized for SWDs, especially for ontologies, is needed by human users as well as software agents and services. At this

stage, human users are expected to be semantic web researchers and developers who are interested in accessing, exploring and querying the RDF and OWL documents found on the Web. Semantic Web search engine to facilitate the development of the Semantic Web, especially the following three activities:

***Finding appropriate ontologies*:** Failing to find a proper ontology always leads to the creation of a new ontology, which is often too customized to be reused. It helps the users to find ontologies containing specified terms, and users may even qualify the type (class or property) of a term. The ranking mechanism sorts ontologies by their popularity.

***Finding instance data*:** In order to help users to integrate
Semantic Web data distributed on the Web, it enables querying SWDs with constraints on the classes and properties used by them.

***Characterizing the Semantic Web*:** By collecting meta data, especially inter-document relations, about the Semantic Web, it reveals interesting structural properties such as “how the Semantic Web is connected", “how ontologies are referenced", and “how an ontology is modified externally".

A system that automatically discovers SWDs, indexes their metadata and answers queries about it. This distinguishes it from other semantic Web repositories and query systems in literature. Ontology based annotation systems, such as SHOE [14], Ontobroker [9], WebKB [15], QuizRDF [8] and CREAM [11], focus on annotating online documents. Their document indexes are based on the annotations but not the entire document, and they use their own ontologies which may not suit for Semantic Web documents. It is notable that CREAM [11] had indexed `proper reference' and `relational metadata'. Ontology repositories, such as DAML Ontology Library [1], SemWebCentral [4] and Schema Web [2], do not automatically discover semantic Web documents but rather require people to submit URLs. They only collect ontologies which constitute a small portion of the Semantic Web. It simply store the entire RDF documents. Some Semantic Web browsers are introduced. Ontaria [5] is a searchable and browsable directory of RDF documents developed by the W3C; It also does not automatically discover SWDs and stores the full RDF graphs. Semantic Web Search [3] indexes individuals of well-known classes (e.g. foaf:Person, rss:Item). Search engine is to design a system that will scale up to handle hundreds of documents.

Semantic Web Documents

Semantic Web languages based on RDF (e.g., RDFS 2, DAML+OIL 3, and OWL 4) allow one to make statements that define general terms (classes and properties), extend the definition of terms, create individuals and to make assertions about terms and individuals already defined or created. A Semantic Web Document (SWD) to be a document in a semantic Web language that is online and accessible to web users and software agents. Similar to a document in IR, a SWD is an atomic information

exchange object in the Semantic Web. Two kinds of documents are semantic Web ontologies (SWOs) and semantic Web databases (SWDBs). A document is considered as a SWDB when it does not define or extend a significant number of terms. A SWDB can introduce individuals and make assertions about them or make assertions about individuals defined in other SWDs. For example, the SWD http://xmlns.com/foaf/0.1/index.rdf is considered a SWO in that its 466 statements (i.e. triples) define 12 classes and 51 properties but introduces no individuals. The SWD http://umbc.edu/~¯nin/foaf.rdf is considered to be a SWDB since it defines or extends no terms but defines three individuals and makes statements about them. Between these two extremes, some SWDs are intended to be both an ontology that defines a set of terms to be used by others, as well as a useful database of information about a set of individuals. Even a document that is intended as an ontology might define individuals as part of the ontology.

Web Search Engine Architecture

An overview of the WSE architecture with particular focus on the data processing which involves the following steps:

- The crawler gathers data from the Web by traversing the link graph and transforms metadata from HTML documents (e.g. RDFa, GRDDL, or Microformats) and metadata embedded in various file formats (e.g. PDF,PNG, MS Office) into RDF. Search engine uses Multi-crawlers 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.
- Reasoning is implemented to improve the quality of data, create new relationships between entities in the data, and perhaps most importantly to merge data from multiple sources and schemas into a consolidated dataset. Reasoning is used by exploiting OWL [10] and RDFS descriptions of a given domain to infer new knowledge about instances in that domain. This can be done in two steps: first indexer and link analyzer builds a graph of the crawled pages. Link analysis is then performed to calculate authoritativeness of web pages.
- It supports SPARQL [11], a W3C Recommendation for an RDF query language. The index structure comprises a complete index on quadruples [12] with keyword search functionality based on a standard inverted index. The index and query processing components can be distributed across a number of machines [13]. The process of collecting and preparing data to allow for the provisioning of query and navigation services is illustrated in Figure 1.

Searcher

This component is responsible for searching and retrieving relevant results. First query analyzer performs mapping of query terms as well as query expansion using an ontology. This component is responsible too for maintaining user log and keeping

track of user search history. Search agent retrieves relevant results from resources database. Retrieved results are then passed to ranking module to be ranked.

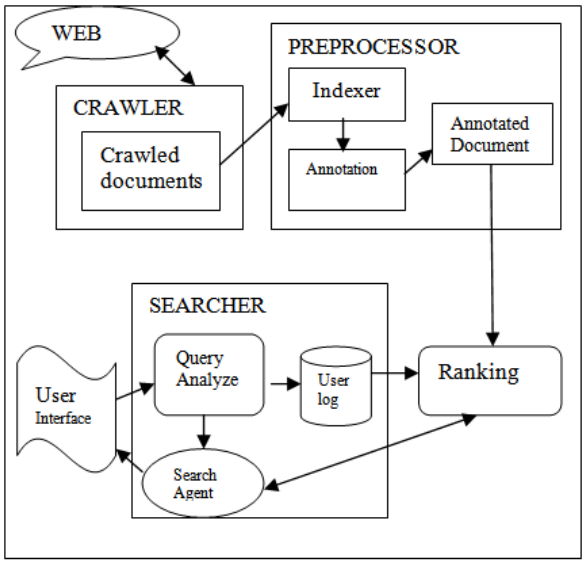

Figure 1: Web Search Engine Architecture.

Ranking module

This module is responsible for ranking the retrieved results. Two factors contribute to the score. The first one is the page authoritativeness which is calculated during the preprocessing phase using link analysis techniques. The second is the relevancy of resource content to query terms which depends on content analysis. Analyzing user's search history can result in a value that represent user's interests in a particular query term. The final ranking score is the combination of these factors.

SWD Metadata

SWD metadata is collected to make SWD search more efficient and effective. It is derived from the content of SWD as well as the relations among SWDs. In Web Search Engine identify three category,(i) basic metadata, which considers the syntactic and semantic features of a SWD (ii) Ontological Indexing and (iii) Ranking measure.

Basic metadata

The basic metadata about a SWD falls in three categories:

Language feature, RDF statistics and ontology annotation.

Language feature refers to the properties describing the syntactic or semantic features of a SWD. Search engine captures the following features:

Encoding
It shows the syntactic encoding of a SWD.
There are three existing encodings, namely “RDF/XML”, “N-TRIPLE” and”N3”.

Language
It shows the language used by a SWD. Search engine considers the usage of four meta level languages, namely “OWL”, “DAML+OIL”, “RDFS”, and “RDF”.

OWL Species
It shows the language species of a SWD written in OWL. There are three possible species, namely “OWL-LITE”, “OWL-DL”, and”OWL-FULL”.

RDF statistics refers to the properties summarizing node distribution of the RDF graph of a SWD. In an RDF graph, a node is recognized as a class if it is not an anonymous node and it is an instance of rdfs:Class; similarly, a node is a property if it is not an anonymous node and it is an instance of rdf:Property; an individual is a node which is an instance of any user defined class. Let foo be a SWD. By parsing foo into an RDF graph .It may get RDF statistics about foo. Let C(foo); P(foo); I(foo) be the set of classes, properties and individuals defined in the SWD foo respectively. The ontology-ratio R(foo) is calculated by equation (1). The value of ontology-ratio ranges from 0 to 1, where “0” implies that foo is a pure SWDB and “1" implies that foo is a pure SWO.

$$R(foo) = \frac{|C(foo)| + |P(foo)|}{|C(foo)| + |P(foo)| + |I(foo)|} \quad (1)$$

Ontology annotation refers to the properties that describe a SWD as ontology. A SWD has an instance of OWL:Ontology, Swoogle records its properties as the following:

1. label. i.e. rdfs:label
2. comment. i.e. rdfs:comment
3. versionInfo. i.e. owl:versionInfo and daml:versionInfo

Ontological Indexing
This system use the classical vector space model to index documents in this Search system. Given a document dj , it is represented by a vector

$$\vec{dj} = (w1,j\ ,w2,j\ ,\ldots..,\ wm,j,\ c1,j\ ,\ c2,j\ ,\ldots\ ,\ cn,j) \quad (2)$$

where m is the total number of index keywords in the system, n is the total number of index concepts in the system, wi,j represents the keyword wi’s weight in document dj , and ci,j represents the concept ci’s weight in document dj ,For each keyword wi, its weight wi,j is calculated using the traditional tf=idf measure

$$w_{i;j} = freq_{i;j} \times \log \frac{N}{n_i} \quad (3)$$

where freqi,j represents wi's frequency in dj , N is the total number of documents, and ni is the number of documents where the keyword wi appears. For each concept ci, we use a simple method to determine its weight ci,j . If the concept ci is specified in the document dj , its weight ci,j is 1, else its weight is 0. This approach is different from the way of page rank-like algorithms to process conceptual information.

The use of N-Gram is particularly important to this approach because of the treatment of URIrefs as terms. Given a set of keywords defining a search, it want to match documents that have URIrefs containing those keywords. For example, consider a search for ontologies for time. The search keywords might be time temporal interval point before after during day month year eventually calendar clock durations end begin zone. Candidate matches might include documents containing URI refs such as:

http://foo.com/timeont.owl#timeInterval

http://foo.com/timeont.owl#CalendarClockInterval

http://purl.org/upper/temporal/t13.owl#timeThing

Clearly, exact matching based on words only would miss these documents (based on the URI refs given).N-Grams would find a number of matches. A custom indexing and retrieval engine we built for the Carrot2 distributed IR system [7]. It can be made to use either n-grams or words, and employs a TF/IDF model with a standard cosine similarity metric. It enhanced to process RDF documents using either character level n-grams computed over the RDF source or to process the URI refs in the document as indexible tokens.

Ranking Measure

The document vector dj and the query vector q, the similarity measure of a document dj to the query q is computed as:

$$sim(d_j, q) = \frac{|d_j| . |q|}{|d_j| \times |q|} \quad (4)$$

This formula is a classical measure used in the vector pace model to calculate a document's similarity to a query. A given document A, A's Page Rank is computed by equation :

$$PR(A) = PRdirect(A) + PRlink(A) \quad (5)$$

$$PRdirect(A) = (1 ¡ d) \quad (6)$$

$$PRlink(A) = d\,(\frac{PR(T1)}{C(T1)} + ::: + \frac{PR(Tn)}{C(Tn)}) \quad (7)$$

where T1; : : : ; Tn are web documents that link to A; C(Ti) is the total outlinks of Ti; and d is a damping factor, The intuition of PageRank is to measure the probability that a random surfer will visit a page. This equation captures the probability that a user will arrive at a given page either by directly addressing it, or by following one of the links pointing to it. Results are ranked according to a final score that represents a combination of two different factors:

The first factor is Page authoritativeness which is calculated using link analysis techniques namely Page Rank algorithm. Authoritativeness value is calculated during the preprocessing phase. The second factor is content relevancy. Query terms in correlation with the weighted annotations are used to calculate query relevancy to each document individually. During ranking stage,[4-6] weights are assigned to terms by analyzing user log and usage data against query terms. The frequencies assigned to profile keywords are significant since they express the rate of user interests. The weighting step starts from these frequencies to calculate profile query term weights. Calculating the weights of the initial query terms is performed by pointing out the highest frequency number and dividing each frequency number by this highest number.

$$PF_{(j,u)=} \frac{\sum sj(u)}{\sum sj_{(u)}} \tag{8}$$

Where:

- $s_j(u)$ is the frequency of term j in user search history.
- $s_k(u)$ is the entire number of terms appeared in user search history

Finally, ranking module calculates the final score using weights calculated from link analysis, weighted annotation

$$Sem_{(i,j,u)} = W_{i,j} + PF_{j,u} \tag{9}$$

$$Score_{(i,j,u)} = \Sigma_{j,q}\, sem_{(i,j,u)} + PR(A_{(i)}) \tag{10}$$

Where

- sem(i,j,u): the similarity between document i and query term j for user u
- score(i,q,u) is the final weight assigned to document i against query q for user u. Ranking module then passes results back to search agent which in turn passes them to user interface.

Conclusion and Future Work

In this paper, a general framework for keyword based Semantic Web Search Engine; It is a crawler-based search engine to traverse both traditional as well as semantic Web. A prototype crawler-based indexing and retrieval system for the Semantic Web Documents. It runs multiple crawler to discover SWDs through meta-search and following links, analyzes SWDs and produce metadata about SWDs as well as the

relations among SWDs. Additionally, user interests and preference are automatically learned from Web usage data and integrated with page authoritativeness and content relevancy to rank final results. It can only search for one type (RDFs) of ontology file, and it only compares the user keywords with the contents of the ontology files wherever they occur. And so it matches indiscriminately the keywords both from concepts and comment fields. During data preprocessing reduces required time. This system describes one of the interesting properties computed for each semantic web document is its rank – a measure of the documents importance on the Semantic Web.

Furthermore, taking resource authoritativeness and content as well as regarding user preferences enhances final result and increases user satisfaction

References

[1] http://www.daml.org/ontologies/, daml ontology library, by daml.
[2] http://www.schemaweb.info/, schema web.
[3] http://www.semanticwebsearch.com/, semantic web search, by intellidimension.
[4] A. Sieg, B. Mobasher, R. Burke, Learning Ontology-Based User Profiles: A SemanticApproach to Personalized Websearch. IEEE INTELLIGENT INFORMATICS BULLETIN, VOL. 8, NO. 1, 2007.
[5] A. Sieg, B. Mobasher, R. Burke.WebSearch Personalization With Personalization with Ontological User Profiles", Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM 2007), 2007.
[6] M. Dudev, S. Elbassuoni, J. Luxenburger, M. Ramanath, G. Weikum , "Personalizing the Search forKnowledge", 2nd International Workshop on PersonalizedAccess, Profile Management, and Context Awareness: Databases (PersDB'08),
[7] R. S. Cost, S. Kallurkar, H. Majithia, C. Nicholas, and Swoogle query result Y. Shi. Integrating distributed information sources with carrot ii. In Proceedings of the 6th International Workshop on Cooperative Information Agents VI, pages 194{201. Springer-Verlag, 2002.
[8] J. Davies, R. Weeks, and U. Krohn. Quizrdf: search technology for thesemantic web. In WWW2002 workshop on RDF and Semantic Web Applications11th International WWW Conference (WWW11),2002.
[9] S. Decker, M. Erdmann, D. Fensel, and R. Studer.Ontobroker: Ontology based access to distributed and semi-structured information. In DS-8, pages 351{369,1999.
[10] Michael K. Smith, Chris Welty, Deborah McGuiness, “OWL Web Ontology Language Guide”, W3C Recommendation, February 10, 2004.http://www.w3.org/TR/owl-guide/
[11] Eric Prud'hommeaux, Andy Seaborne. “SPARQL Query Language for RDF”, W3C Recommendation, January 15, 2008. http://www.w3.org/TR/rdfsparql-query/

[12] Andreas Harth, Stefan Decker. "Optimized Index Structures for Querying RDF from the Web". 3rd Latin American Web Congress, Buenos Aires, Argentina, October 31 to November 2, 2005, pp. 71-80.

[13] Andreas Harth, Juergen Umbrich, Aidan Hogan, Stefan Decker."YARS2: A Federated Repository for Querying Graph Structured Data from the Web". 6th International Semantic Web Conference, Busan, Korea, November 11-15, 2007.

[14] S. Luke, L. Spector, D. Rager, and J. Hendler. Ontology-based web agents. In roceedings of the First International Conference on Autonomous Agents (Agents97), pages 59{66, 1997.

[15] P. Martin and P. Eklund. Embedding knowledge in web documents. In Proceedings of the 8th International World Wide Web Conference (WWW8), pages 324{341, 1999.