Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 676 – 681

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Domain Ontology Driven Fuzzy Semantic Information Retrieval

Remi S[a,*], Varghese S C[a]

[a]Department of Computer Science, Rajagiri School of Engineering & Technology, Kakkanad, Kochi 682 039, India

Abstract

With the exponential growth in web content, the answers provided by traditional search engines by query specific keywords to content has resulted in markedly high recall and low precision. Semantic information retrieval can enhance the relevancy of search results by understanding search intention and the contextual meaning of terms as they are entered by the user. In this paper, a novel method for supporting semantic information retrieval is proposed by building a domain specific ontology. A prototype of a fuzzy semantic search engine is developed and the results are compared with that of a traditional search engine.

Keywords: Domain Ontology;Fuzzy Search;Information Retrieval;Semantic Query Processing;Semantic Web

1. Introduction

With rapidly growing amount of available data in databases and other data repositories, the need for ordinary users to search such information has increased dramatically. Traditionally, to access such resources, users are required to learn structured query languages, such as SQL and XQuery[1]. Search engines on the Web have popularized an alternative unstructured querying mechanism called Keyword Search that is very simple, user-friendly and requires no schema knowledge.

To find relevant information related to a particular subject from the huge data present on internet, an efficient search technique is necessary. The traditional web searching technique, keyword search, which is adopted by many web search engines, have shortcomings which include reduced precision in results, large set of irrelevant search

* Corresponding Author . Tel.: +91-8592022643.
E-mail address: remis1889@gmail.com

doi:10.1016/j.procs.2015.02.122

results which in turn increases the result retrieval time, inability to judge the meaning of the user's query etc[2]. To overcome these problems semantic search is necessary.

Semantic search means searching the web on the basis of meaning of user query. For supporting effective keyword search by producing more relevant search results, it is necessary to understand and interpret the query in the way user wants. Most of the time, the meaning of the query is hidden in the user query itself. For example, interrogative keywords such as why, what, where etc. can help in understanding user intention. Traditional keyword search methods remove such frequently used keywords by including them in the list of stop words. Semantic search can make use of them to retrieve better results.

Ontology is a semantic web tool that is being increasingly used for building the applications for the specific domain. Ontology enables users to capture the semantic of the documents[3]. In this paper, we construct a domain specific ontology which can be used to identify the user search intention and answer the user query with relevant results as opposed to traditional search systems which only performs regular text matching and hence returns irrelevant results.

This paper is organized as follows: we start by surveying the relevant literature in Section 2. Next, we introduce domain specific ontology based semantic search engine in Section 3. Experiments and results are reported in Section 4 and finally, Section 5 concludes the paper.

2. Related Work

Semantic Web technology was proposed by Tim Berners Lee in 2000[4]. According to him “The Semantic Web is not a separate Web but an extension of the current one, in which information is given well defined meaning, better enabling computers and people to work in cooperation".

Various ontology-based semantic search techniques and approaches designed to perform semantic information retrieval have been proposed by the researchers. We studied a number of techniques proposed and implemented for efficient information retrieval and ontology creation. The existing approaches and the systems are not suitable enough to get the relevant information.

Ilyas et al. proposed a Conceptual Architecture for SSE[5] with the main focus on an inference engine. This model provides a complete knowledge base created using a relational database. Due to this knowledge base, they claim that their method showed high performance as the data can be directly queried from it, rather than from the conventional data. However, this architecture assumes that annotated Web data is already at hand which is not true, hence this is a major limitation of this method.

Zhi-Qiang et al.[6] have proposed a semantic web search engine framework based on domain ontology. The architecture consists of a semantic search engine where the web is crawled, and an information extraction algorithm based on ontologies is used to extract information from the crawled Web pages. Further, a ratiocination machine analyzes the user query with the help of semantic reasoning based on ontology. The authors modeled a prototype semantic search engine using Lucene, and their results show an effective and semantic validity with good recall, precision and retrieval rate above 90% for the semantic request within the given ontology. On the other hand, it was observed that the search results are not exactly relevant if the semantics of user query is beyond the range of the constructed ontology. Further, the process of measuring the similarity of ontology in the query process is also not well defined.

A new semantic indexing and search system was proposed by F. Salam[7], which retrieved web documents based on ontology. The proposed system (MIRO) made use of ontologies to exploit the semantic content of documents to better index them and reduce the silence and increase the accuracy of the research. MIRO includes the following three parts: Indexation, Research and Presentation and Ontology enrichment.

A semantic search technique[8] is proposed that considers the type of desired Web resources and the semantic relationships between the resources. A novel ranking model was also presented in which the following three criteria were considered - the number of meaningful semantic path instances, the coverage of keywords and the distinguishability of keyword.

A new framework for semantic expansion search was presented[9] the basis of which involved constructing a domain specific ontology. A semantic annotation algorithm and semantic expansion reasoning algorithm associated with semantic annotation unit and semantic expansion reasoning engine respectively is also proposed. Semantic

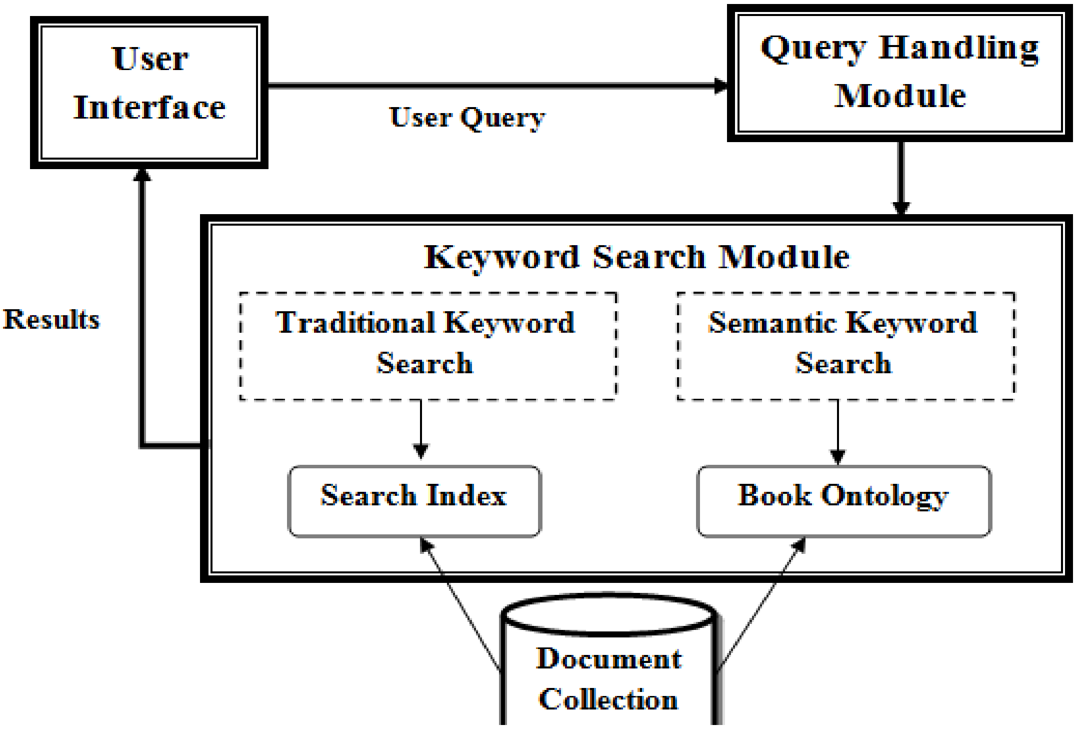

Fig. 1. Overall System Architecture

expansion search uses semantic expansion reasoning algorithm to generate query expansion set for the user input. According to the query expansion set, search is performed and relevant results are returned to user.

An ontology-based information extraction and retrieval system focusing on soccer domain is presented[10] which contains all the aspects of Semantic Web - from information extraction to information retrieval and uses technologies such as OWLDL, inference, rules, RDF repositories and semantic indexing.

Sharma et al.[11] proposed architecture for semantic information retrieval to enhance the relevancy of search results by considering the two factors for calculating the page rank - frequency of the keyword occurred in the web page and associative factor of the same keyword with the meaningful interrogative words which are generally ignored by the other search schemes.

3. Domain Specific Ontology Driven Semantic Search Engine

Traditional keyword search systems use a centralized database for indexing information. They are based on queries from simple keywords. The recall rate is high, but the accuracy is low. This is due to the disambiguation, wrong context, the use of different words (synonyms), more specific words, or more general (hypo-hyperonymic). These systems rarely take into consideration the semantic content of the document to the index. A new approach is required which allows taking into consideration the semantics of the document and focuses on techniques of information retrieval based on ontologies.

3.1. System Architecture

The overall system architecture is shown in Fig. 1. The system consists of four basic modules: Domain Ontology Construction, User Interface, Query Handling and Keyword Search Module.

The user interface in our system is a simple google-like interface that allows the user to input a query with a few keywords which represents the user needs. The input query is submitted to the query handling module which does some preprocessing steps on it. The raw query submitted by the user should be processed before searching by performing several preprocessing tasks such as stop-word elimination, stemming and other application specific tasks. The resulting well-defined system specific query is given to the keyword search module. The output of this module will be a ranked list in order of its relevance to user query which is then returned to the user interface.

Keyword search module forms the core of the proposed system. It consists of two independent modules: Traditional Keyword Search and Semantic Keyword Search.

For performing traditional keyword search, a search index is built from the document collection by performing some preprocessing steps such as tokenization, stop-word removal etc. For each keyword in the user query, it returns results which contain the query keywords exactly. Relevant results are obtained from the data collection using the pre-built index structures. For each keyword entered by the user, its corresponding inverted list is retrieved. If there are multiple keywords in the user query the union of the lists of each keyword is obtained and a combined list is computed. Results returned are ranked in the order of their relevance to the user query by assigning each result a relevance score such as TF-IDF similarity.

In case of semantic search, the user query is converted into SPARQL query and executed against the generated ontology using a SPARQL endpoint. The results from the SPARQL query is processed and displayed to the user through the user interface. Semantic query processing involves converting the plain text user query to a set of concepts defined in the ontology by identifying the user search intention. This is done by performing pattern matching using regular expression in a POS tagged query. Semantic ranking is done based on the search criteria extracted from the semantic query preprocessor. Different category is given a unique weightage specific to that category on a priority basis. The weight of each category is computed with respect to the query criteria. The results are then displayed in a sorted order based on individual result weight. If the user query consists of multiple criteria, the result weights are aggregated and ranked accordingly.

3.2. Constructing an ontology for book domain

Ontology is defined as a "specification of a representational vocabulary for a shared domain of discourse - definitions of classes, relations, functions, and other objects"[12]. It shows the hierarchical relationship between different classes and their subclasses in a graphical pattern. A domain specific ontology can be built from the tokens produced in the preprocessing step so as to facilitate efficient semantic search.

The book ontology used in the proposed system is constructed using the steps given by Natalya et al.[13]. The first step is to collect important concepts related to books such as category, book title, author, genre, publisher, published date, length etc. In the second step, identify the classes and their subclasses. In the next step, identify the properties of those classes and subclasses. After defining the classes, object properties, data properties and other relationships, the ontology is populated with data. Ontology population involves instantiating all the defined classes and assigning values for data and object properties with regards to the constraints imposed on them. The values for populating the ontology is obtained from the tokens stored in the database after document processing. Other rules are inferred using a reasoned and the final ontology is developed and saved. Finally, it is exported in RDF or OWL data format.

3.3. Fuzzy search

Consider the case where the user wants to search for all books by *Christopher Buckley*. Traditional search systems require the user to provide this name exactly, in order to find relevant results. If the user types in *Christofer Buckley* instead of *Christopher Buckley*, then the system will be unable to find the required records. Similarly, traditional search system also fails if the user wants to search by a combination of initials and first name or last name. Thus, if the user types in *C. Buckley* or *Christopher B.* instead of the full name as specified in the index, the system fails to retrieve results. By incorporating fuzzy search, the user can search for books by author without providing author full name. Fuzzy search or approximate search refers to the technique of discovering records that match a keyword approximately (rather than exactly). Similarity between two strings can be quantified using edit distance which is calculated by counting the minimum number of operations required to transform one string into the other.

Formally, the edit distance between two strings s_1 and s_2, denoted by $ed(s_1, s_2)$, is the minimum number of single-character edit operations (i.e., insertion, deletion, and substitution) needed to transform s_1 to s_2[14]. For example, *ed*(“brown”, “brawn”) = 1. Fuzzy search is more powerful than exact search and can be used to locate individuals based on incomplete or partially inaccurate identifying information in the user search query. In this paper, we use a commonly used variant of edit distance called Levenshtein distance to facilitate fuzzy semantic search.

4. Implementation Details and Results

We used Protégé - 4.3[15] for building the book ontology and Pellet reasoner for checking the consistency of the Ontology. OWL (Web Ontology Language) is used to create the book ontology which is populated using the data from the relational database using PHP scripts. SPARQL is used to query information from the built ontology using the SPARQL interface provided by the python library *rdflib*.

Our corpus consists of around 2000 documents, related to books, authors and publisher, retrieved from Wikipedia. For testing the proposed system, book pages from Wikipedia in XML format were parsed to obtain the infoboxes which contains relevant details such as book title, author name, publisher, publication date, genre etc. and are stored in a relational database using PHP. The author and publisher details are extracted from the respective articles related to the authors and publishers from wikipedia.

Ten queries that contain various topics and consist of few terms were randomly determined as shown in Table. 1. After each run of the query, the results retrieved were evaluated using binary human relevance judgment and with this, every result was classified as relevant or non-relevant.

Table 1. Test queries.

Label	Type	Query
Q1	Author	Books written by *Christopher Buckley*
Q2	Publisher	Books published by *Random House*
Q3	Book, Publisher	Books by publisher of *Angels & Demons*
Q4	Book, Author	Books by author of *Deep Wizardry*
Q5	Genre	*Suspense* Novels
Q6	Category	*Non Fiction* Books
Q7	Author	Books by *Robert Jordan*
Q8	Book	*And Then There Were None*
Q9	Author	*Stephen King*
Q10	Publisher	*Harper Collins*

Precision and Recall are the main evaluation criteria used for comparing system performance. Precision, one of the most commonly used metrics in IR, measures how precisely the system picks the related documents among all documents. It is the proportion of the related documents in the retrieved documents (true positives) to the total number of retrieved documents. The value of precision lies between 0 and 1 and higher precision value indicates better retrieval performance.

Recall, which is another widely used IR metric, is the proportion of the retrieved related documents to the total number of related documents that should have been retrieved. The value of recall also lies between 0 and 1 and higher recall value indicates better retrieval performance. When used together, precision and recall give a solid idea about the retrieval performance.

For comparison purpose, we also developed a traditional keyword search system with the same dataset using inverted index as its core index structure and tf-idf ranking scheme to rank the results. Precision and recall values are computed for ten sample queries, given in Table. 1, with both traditional and semantic search systems and the results are tabulated in Table. 2. The relevancy of the retrieved results is evaluated manually. The results show that the system achieves an increase in average precision 78 % and 13 % increase in average recall.

Table 2. Precision and Recall for the test queries in Table. 1.

Query	Traditional keyword search		Semantic keyword search	
	Precision	Recall	Precision	Recall
Q1	0.01	1	1	1
Q2	0.25	0.4	1	1
Q3	0.04	1	1	1
Q4	0.02	1	1	1
Q5	0.75	0.3	1	1
Q6	0.01	1	1	1
Q7	0.02	1	1	1
Q8	0.03	1	1	1
Q9	0.39	1	1	1
Q10	0.62	1	0.7	1
Average	0.21	0.87	0.97	1

5. Conclusion

The answers provided by traditional search engines have resulted in high recall and low precision. Semantic information retrieval can enhance the relevancy of search results by understanding search intention and the contextual meaning of terms as they are entered by the user. A novel method for supporting fuzzy semantic information retrieval using ontology is proposed. The proposed system is an attempt to retrieve relevant results in the book domain using domain specific knowledge captured in the form of OWL ontology. Experimental results show that the system retrieves the documents which would have been missed and avoids retrieving documents which are irrelevant despite the presence of the keyword. It thus improves the precision and recall substantially.

References

1. Li Y, Yu C, Jagadish HV. Schema-Free XQuery. *Proceedings of International Conference on Very Large Data Bases*; 2004. p. 72-83.
2. Sharma R, Kandpal A, Bhakuni P, Chauhan R, Goudar RH, Tyagi A. Web page indexing through page ranking for effective semantic search. *Intelligent Systems and Control (ISCO)*; 2013. p. 389-392.
3. Sharma R, Kandpal A, Bhakuni P, Chauhan R, Goudar RH, Tyagi A. Domain Ontology based Semantic Search for Efficient Information Retrieval through Automatic Query Expansion. *ISSP*; 2013. p. 397-402.
4. Berners-Lee T, Hendler J, Lassila O. The semantic web, *Scientific American* 284:5; 2001. p. 28-37.
5. Ilyas QM, Kai YZ, Talib MA. A Conceptual Architecture for Semantic Search Engine, *IEEE*; 2004. p. 605-610.
6. Zhi-Qiang DU, Jing HU, Hong-Xia YI, Jin-Zhu HU. The Research of the Semantic Search Engine based on the Ontology. *WiCom IEEE;* 2007. p. 360-363
7. Salam F. New Semantic Indexing and Search System Based on Ontology. *EIDWT*; 2013. p. 313-318.
8. Li J, Min J, Chung C. An effective semantic search technique using ontology. *Proceedings of the 18th international conference on WWW;* 2009. p. 1057-1058.
9. Zou G, Zhang B, Gan Y, Zhang J. An Ontology-Based Methodology for Semantic Expansion Search, *FSK,*; 2008. p. 453-457.
10. Kara S, Alan O, Sabuncu O, Akpinar S. An ontology-based retrieval system using semantic indexing, *ICDEW IEEE*; 2010. p. 197-202.
11. Chen Y, Wang W, Liu Z, Lin X. Keyword Search on Structured and Semi-Structured Data. *Proceedings ACM SIGMOD International Conference on Management of Data*; 2009. p. 1005-1010.
12. Gruber TR. A Translation Approach to Portable Ontology Specifications, *Knowledge Acquisition:* 5:2; 1993. p. 199-220.
13. Natalya F, Deborah L, Mc. Guinness. Ontology Development 101: A Guide to Creating Your First Ontology, *Stanford University*; 2000.
14. Li G, Wang J, Li C, Feng J. Supporting Efficient Top-k Queries in Type-Ahead Search. *SIGIR*; 2012. p. 355-364.
15. Protege: An Ontology Editor. Technical report, July 2014. Available at http://protege.stanford.edu.