Expert Systems with Applications 41 (2014) 8225–8233

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An architectural design for effective information retrieval in semantic web

M. Thangaraj [a], G. Sujatha [b,*]

[a] Madurai Kamaraj University, Madurai 625 021, India
[b] Sri Meenakshi Govt. Arts College for Women(A), Madurai 625 002, India

ARTICLE INFO

Article history:
Available online 19 July 2014

Keywords:
Information retrieval
Semantic web
Semantic search
Ontology
Semantic query

ABSTRACT

The current web IR system retrieves relevant information only based on the keywords which is inadequate for that vast amount of data. It provides limited capabilities to capture the concepts of the user needs and the relation between the keywords. These limitations lead to the idea of the user conceptual search which includes concepts and meanings. This study deals with the Semantic Based Information Retrieval System for a semantic web search and presented with an improved algorithm to retrieve the information in a more efficient way.

This architecture takes as input a list of plain keywords provided by the user and the query is converted into semantic query. This conversion is carried out with the help of the domain concepts of the pre-existing domain ontologies and a third party thesaurus and discover semantic relationship between them in runtime. The relevant information for the semantic query is retrieved and ranked according to the relevancy with the help of an improved algorithm. The performance analysis shows that the proposed system can improve the accuracy and effectiveness for retrieving relevant web documents compared to the existing systems.

1. Introduction

1.1. Motivation

The World Wide Web serves as a huge wide distributed global information center for many information services. By now the size of the web is billions of websites and is still growing rapidly. But to get an exact requirement a normal user often spends a lot of time. In order to present the relevant results from this voluminous data to the user, some new methods should be derived to filter the results. The current information technology from the web is mostly based on the keywords. It provides limited capabilities to capture the concept of the user requirement. To solve the limitations of the keyword based search the idea of semantic search is introduced in the field of information retrieval (IR). Information retrieval is the science of searching for documents, information within the documents as well as that of relational database and the World Wide Web. IR also deals with representing, storing and organizing the content.

Semantic search has been presented in the IR field since the early eighties (Croft, 1986). The use of ontologies with keyword based search is one of the motivations of the semantic web (SW). The semantic web "targets to build an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" (Berners-Lee, Hendler, & Lassila, 2001). Fig. 1 shows the layers of the semantic web as suggested by Berners-Lee.

The bottom layer contains technologies that provide basics for the SW. *Uniform resource identifiers* (URIs) provide a standard way to refer to entities, while *Unicode* is a standard for exchanging symbols. The *Extensible Markup Language* (XML) fixes a notation for describing labelled trees, and XML Schema allows the definition of grammars for valid XML documents. XML documents can refer to different *namespaces* to make explicit the context (and therefore meaning) of different tags. The middle layer contains technologies to enable building SW applications. Resource Description Framework (RDF) is a framework for creating statements in the form of resources, properties and statements as triples. RDF schema provides a basic vocabulary of RDF. Web Ontology Language describes semantics of RDF statements. SPARQL is RDF Query language. Top layer contain just ideas that should be implemented in order to realize SW. Cryptography, Trust and Proof is to ensure that the

* Corresponding author.
E-mail addresses: thangarajmku@yahoo.com (M. Thangaraj), sujisekar05@rediffmail.com (G. Sujatha).

http://dx.doi.org/10.1016/j.eswa.2014.07.017
0957-4174/

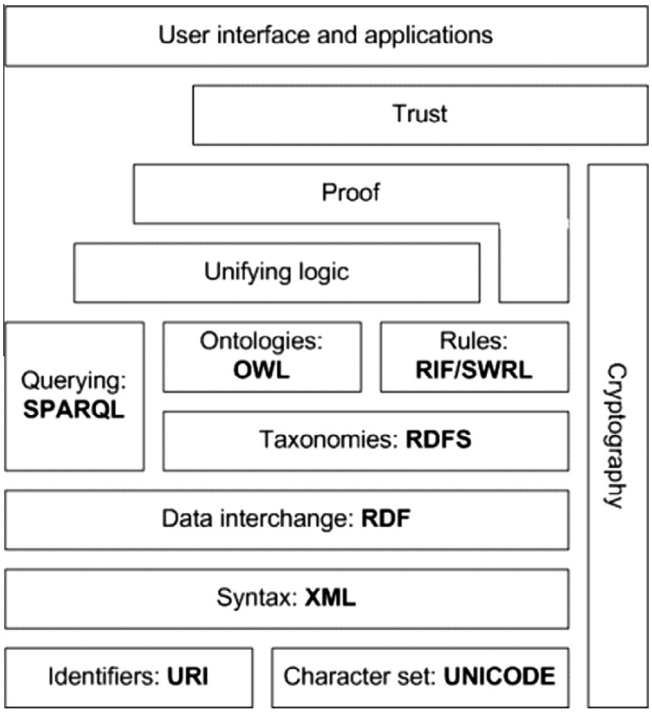

Fig. 1. Semantic web architecture.

SW statements are from trusted source. User interface is the final layer that will enable humans to use SW applications.

Some of the main goals of the semantic web are Semantic Query Processing and discovering the semantic information available in the unstructured web information with the help of domain ontologies. The IR from the semantic web combines the fast-developing research areas information retrieval, semantic web, and Web content Mining.

The various problems associated with the unstructured web pages are identified as follows. (i) Web pages are far complex than that of any traditional document collection. (ii) The web is a highly dynamic information source. (iii) The web serves a broad diversity of user communities. (iv) Only a small portion of the information on the web is truly relevant or useful. These challenges have promoted research into effective and efficient discovery and uses of resources on the internet. There are extensive research activities on the construction and use of semantic web which is nothing but the structure of semantic meaning of the content of the web pages. Web document classification by web mining will help in building the ontology for the semantic web with the automatic extraction of the semantic meaning of web pages.

In order to address this issues, the Semantic Based Information Retrieval System (SBIRS) mechanism for SW is proposed. This architecture handles the semantic indexing, extraction, extensions of query and matching of content semantics to achieve the following objectives. (i) Analyze and determine the semantic feature of the content by means of semantic annotation. (ii) Analyze the user's query and extend it to semantic query using link extraction, ontology and thesaurus. (iii) Match the semantic query with the semantic content using a semantic indexing structure. (iv) Arrange the retrieved results in the order of their relevancy to the query using proposed dynamic ranking algorithm. This architecture eliminates the problems of traditional keyword search and enables the user to retrieve the concept oriented relevant results for any domain.

The significance of the framework is to improve search accuracy by understanding searcher intent and the contextual meaning of terms as they appear in the searchable data space. SBIRS also has a few aspects that distinguish it from other related work. Unlike typical search algorithms this framework is based on keyword-to-concept mapping with an improved semantic indexing structure and searching technique. The proposed dynamic ranking algorithm presents the results in the order of their relevance for the expanded semantic query.

1.2. Research contributions

Contributions of this research fall into the following categories.

- Clear knowledge of the semantic search, possibilities of semantic enhancements in the IR models.
- Definition and implementation of a semantic retrieval model with generic domain ontology.
- Creation of an improved semantic indexing structure.
- Implementation of a dynamic ranking algorithm
- Investigation of the feasibility of semantic retrieval in cloud environment.
- Checking the feasibility of semantic image retrieval.

1.3. Structure of the paper

The rest of the paper is organized as follows. An overview of related work is given in Section 2. In Section 3 the working mechanism of the proposed architecture is explained. Section 4 elucidates the retrieval and the ranking algorithms. The performance evaluation is given in Section 5. In Section 6 the main achievements and the tasks that remain is discussed.

2. Related work

The unsolved problems of current search engines have led to the development of semantic web search system (Yi Jin & Hongwei Lin., 2008). Conceptual search has been the motivation of a large body of research in the IR field long before the semantic web vision emerged (Jo rvelin, Kekalainen, & Niemi, 2001). "SemSearch" (Yuangui Lei & Enrico Motta, 2006) is a layered architecture that separates end users from the back-end heterogeneous semantic data repositories. "SemSearch" accepts keywords as input and delivers results which are closely relevant to the user keywords in terms of semantic relations. The SBIRS compliments SemSearch with a ranking algorithm designed specifically for an ontology-based information retrieval model with a semantic indexing structure based on annotation weighing techniques.

The inherited relationships between the keywords are analyzed in terms of concepts in "Ontolook" (Li, Wang, & Huang, 2007). From this concepts and relations a concept-relation graph is formed which is used to eliminate the less ranked arcs. It also creates a property-keyword candidate set and sent it to the web page database to get a retrieved result set for the users. The efficiency of this approach is limited by lack of ranking technology. This motivates a relation based page ranking algorithm for semantic web search (Lamberti, Sanna, & Demartini, 2009). The ranking technology is based on the estimate of the probability that keywords/concepts within an annotated page are linked one with another in a way that is the same to the one in the user's mind at the time of submitting the query. The probability is measured using a graph-based description of ontology, user query and the annotated page. In these approaches further efforts are requested for future semantic web repositories based on multiple ontologies and better ranking. By building upon a dynamic ontology our model supports multiple domains with semantic dynamic ranking.

An alternate model using ontology is given as an adaptation of vector space model for ontology based information retrieval (Pablo Castells & David Vallet, 2007). This model handles an ontology-based scheme for the semi-automatic annotation of documents and retrieval system. The classical vector space model (Salton and McGill, 1983) is combined with an annotation weighting algorithm and a ranking algorithm. The assumption here is a knowledge base (KB) which has been built and associated to the document base with the help of one or more domain ontologies which describe concepts appearing in the domain text. The documents are then annotated and semantically indexed with the help of the KB. The query is accepted in RDQL form and executed against the knowledge base which returns a list of instances that satisfy the query. This annotation weighting scheme is not considering the different relevance of document fields like title etc. It is addressed in SBIRS by boosting the weight of the query variables in condition on web document tags according to the importance of fields. This work is extended in Miriam Fern'andez, Vanesa L'opez, and Pablo Castells (2011), by accepting the input in a formal SPARQL query and later modified in the form of natural language query. This proposal has investigated the practical feasibility of applying semantic search modes in the web environment. Our proposal considers a large scale of heterogeneous web environment.

An improved version is discussed in "The Google Similarity Distance" (Rudi Cilibrasi and Paul Vitan'nyi, 2007) which deals with words and phrases that acquire meaning from the way they are used in society from their relative semantics to other words and phrases. The similarity is based on information distance and kolmogorov complexity. These works are concerned with the similarities between two entities in the ontology. Our approach concerned with the similarity between the query and the web documents.

The cluster based approach for information retrieval provides features 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. Users can zoom in on smaller clusters, and then drill down through subgroups (Ramesh Singh & Aman Arora, 2010). Whereas this work is concerned with the query expansion, SBIRS is concerned with starting from query expansion to retrieve ranked results.

A Crawler-based indexing and information retrieval system for the semantic web Swoogle (Li Dong et al., 2004) extracts meta data for each discovered document, and computes relations between documents. The ontology rank is computed as a measure of the importance of a semantic web Document. Swoogle is improved by adding user preferences and interests to provide user a set of personalized results. Swoogle is strictly for semantic web documents whereas SBIRS approach converts web documents into semantic web documents.

A search engine that uses several mapped RDF ontologies for concept based text indexing is discussed in Jacob Köhler and Michael Specht (2006). For any information retrieval system ranking algorithm is defined with certain metrics. The variety of relevance ranking metrics are discussed and analyzed in Xavier Ochoa (2008). It proposes a set of metrics to estimate the personal, topical and situational relevance dimensions. These metrics are calculated mainly from contextual information and usage and do not require any explicit information from users. Our work moves from the consideration above and relies on the assumption that for providing effective ranking the SBIRS logic makes use of the underlying ontology and of the web page to be ranked in order to compute the corresponding relevance score.

A semantic-based approach to content annotation and abstraction for content management is proposed in Hui-Chuan, Chen, and Chen (2009). In this approach a semantic-driven content environment which features a high interoperability of content can be constructed for bridging the semantic gaps for the customer and the content author to increase the efficiency of content management. This work is improved based on the semantics consisting of elements like subject, predicate, and object (Ming-Yen, Chu, & Chen, 2010). In this work a semantic pattern expression capable of representing content semantic features has been designed to represent human semantics with topic-to-topic associations and to replace keywords as the input of the information retrieval system. This architecture contains the core technologies such as Semantic determination and extraction, Semantic Extension, Semantic Pattern and matching. Compared to these approaches, the proposed architecture considers web documents and a much more detailed, densely populated conceptual space in the form of ontology based knowledge base and thesaurus instead of a topic map.

3. Proposed SBIRS architecture

The SBIRS architecture for information retrieval from semantic web uses conceptual representations of content beyond plain keywords as knowledge bases and provides conceptual representations of user needs. This architecture handles the concept representations of the content, query extensions, matching the semantics, extraction of the relevant results in the order of relevancy with the help of the following components.

- Crawler
- Preprocessor
- Semantic annotator
- Semantic indexer
- Semantic query converter
- Semantic content retriever
- Semantic ranker

These components are grouped under different layers of the architecture. The Physical Data Layer creates web database with the components crawler, preprocessor. The Semantic Annotation Layer creates knowledge base with semantic annotator and indexer. Semantic Matching Layer performs matching between semantic content and the semantic query. The retrieval layer ranks the retrieved results and is submitted to the Application Layer. The overall architecture with the above said components are given in the Fig. 2.

3.1. Crawler

The crawler collects the web pages from different domains. The collected web pages are stored to a web database for the use of future retrieving URLs and corresponding web pages. The result of the crawler is sent to the preprocessor for getting the pure content from this unstructured web documents.

3.2. Pre-processor

The unstructured web documents are pre-processed before matching into ontology concepts. Using HTML parser the meaningless HTML tags are removed. After extracting text from the web documents the less meaningful words known as stop words like neuter pronouns, articles, and symbols are removed. The last process is stemming to convert into root words. Stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form generally a written word form. While processing documents, this preprocessor will filter images, audio, video and other information formats, and will identify and

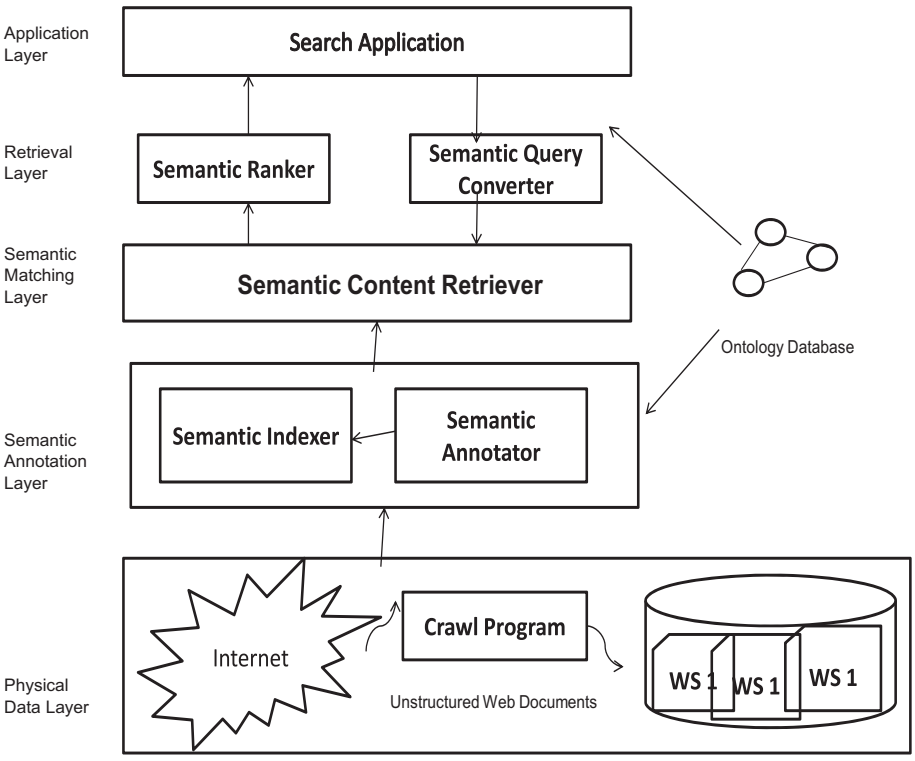

Fig. 2. Semantic Based Information Retrieval System (SBIRS) architecture.

eliminate the noise content. The same process is repeated for the user query on stop words and derived words.

3.3. Semantic annotator

Semantic annotation is a type of meta data generation and usage schema used to extend the existing information access methods. The annotation scheme used here is based on the concepts of the particular predefined domain ontology and the meanings of the phrases as semantic entities. Those entities can be coupled with formal descriptions and thus provide more semantics and connectivity to the web database.

With the help of the domain ontologies, the web database created by the crawler module is converted into knowledge base. That means the concept of the web page is matched with the context of the concept in the predefined ontology.

As a result of semantic annotation process, the web documents are associated with their corresponding keywords and semantic entities (Concepts and Meanings). But in the case of keyword based system the web documents are associated with the keywords alone.

3.4. Semantic indexer

The annotated web documents of the knowledge base resulted from semantic annotator are indexed with the semantic entities. The mapping score which indicates how good a web document is mapped to an ontological concept is computed. And hence the indexer creates a weighted semantic annotation/indexing. Scores are computed by an adaptation of improved TF-IDF algorithm.

The weight is a function of term frequency of the keyword [*tf* (W)], term frequency of concept [*tf* (C)], Tag based keyword frequency [*tagf* (W)], and Tag based concept frequency [*tagf* (C)] and normalization factors. For the tag based frequency specific tags of the HTML file such as Head, Title, Meta, Description, Anchor tags are considered. The importance will be given for the presence of keyword and/or concept in the URL of the web page. The weights thus calculated are used later for ranking.

3.5. Semantic query converter

The plain keywords entered by the user is expanded and converted into semantic query in three different ways. (i) By matching the concepts in the domain ontology. (ii) By using the links of the websites with the help of an automatic link extraction algorithm. (iii) By using the thesaurus. The Extended semantic query is presented to the user as ontology suggestion. The user by making

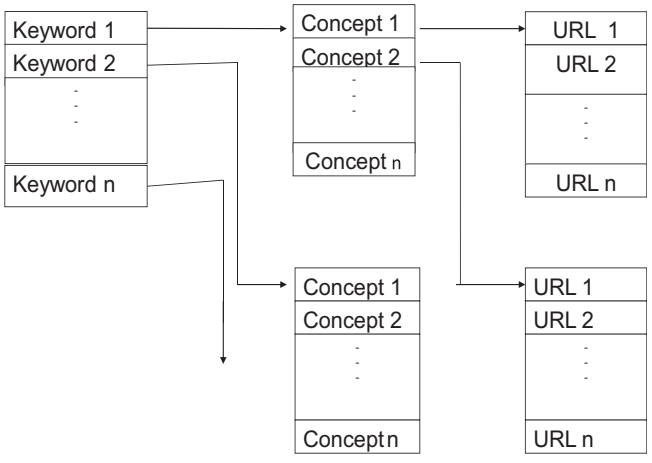

Fig. 3. Proposed mapping of keywords with concepts of ontology.

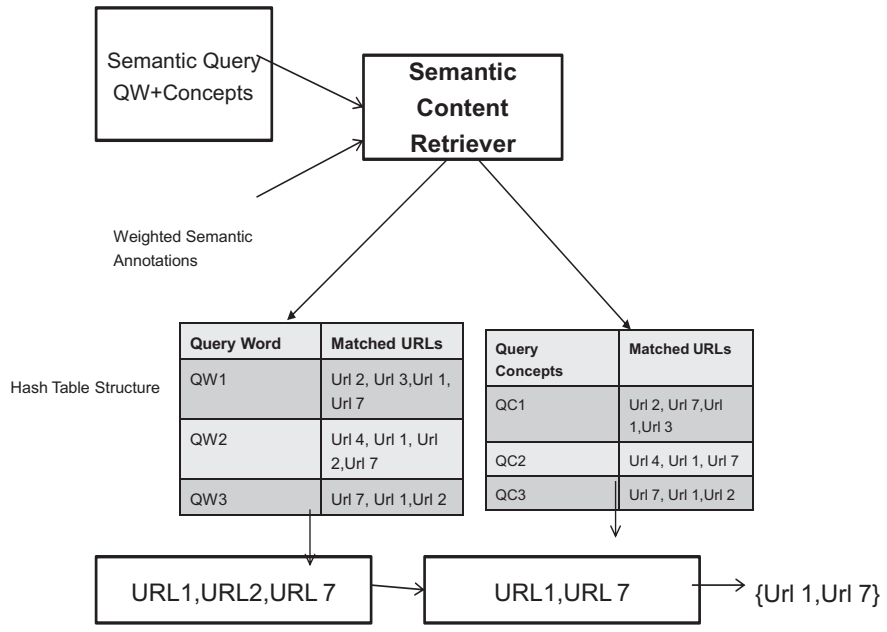

Fig. 4. Process of semantic content retriever.

use of the suggestion selects his concept of searching and submits it to the semantic content retriever. The Query extension with the concepts and matching of web documents is given in Fig. 3.

3.6. Semantic content retriever

This component concerns with identifying and submitting the most approximate content to querists by matching in the semantic content for the query semantic patterns, and it covers the following steps: The semantic query from the semantic query converter is matched here with the semantically indexed web content. The retrieved content should be matched with the two parts (keyword and concept) of the semantic query.

The final retrieval list is the intersection between the set of the web documents containing the keyword and the semantic entity/contextual meaning. To retrieve the intersecting list a hash table structure which is having two columns is used.

The first column has the Query words of the extended query and the second column has the list of web documents that matches with that Query words. This process is depicted in the Fig. 4. The resultant list will be the intersection of the web documents that are stored in the second column.

3.7. Semantic ranker

The retrieved list of the previous module is ranked with the help of the semantic weights. The relevancy is measured with the weights calculated by the improved dynamic ranking algorithm. The weight is a function of term frequency, collection frequency and also some normalization factors. The term frequency is the local weighting factor which reflects the importance of the term within a particular document. The global weighting factor considers the importance of a term within the entire collection of documents known as document frequency (*df*). Next the inverse document frequency (*idf*) which relates the document frequency to the total number of documents in the collection (*N*) is computed. In contrast with the keyword based system these values will be calculated for the keyword as well as for the semantic entities.

$$idf = log\frac{N}{df} \tag{1}$$

The weight of a term w_j or concept c_j in a document D_i is defined as a combination of *tf* and *idf*.

$$wt_{ij}(w_{ij}) = tf(w_{ij}) * idf(w_j) \tag{2}$$

$$wt_{ij}(c_j) = tf(c_{ij}) * idf(c_j) \tag{3}$$

Now the similarity coefficient (*sc*) between the query and the web document is defined by the dot product of weights of the corresponding words and semantic entities of the semantic query (*n*) and the web document.

$$sc(Q, D_i) = \sum_{j=1}^{t} wt_{qj} * wt_{ij} \tag{4}$$

Based on the similarity values the most relevant results identified by a threshold value are presented to the user. In the improved algorithm the weight depends on the two main factors. One is the quoted frequency of the keywords in the web documents and another one is the semantic entities in the content of the web pages. When there is no semantic entity the retrieval is nothing but the keyword based.

The improved algorithm based on TF-IDF algorithm can fit both traditional web and semantic web making the IR more accurate and promote the efficiency and the precision of traditional web search and semantic web search.

4. Algorithms

This section presents the various algorithms used in the different components of the semantic search architecture proposed.

Algorithm: Semantic Annotation and Indexing
Input: Set of Web Documents (D) in a particular domain (N)
: Set of concepts from domain ontology
Output: Semantic Content Knowledge Base with its score
Parameters: ***N***-Total Number of Web documents, D-Set of Web documents,
S-Stop words, w_{ij} – jth word in ith document,
m-Number of Keywords in a Webdocument,
tf-Term Frequency, *tagf*-Frequency of terms in HTML tags,
c-Total number of Concepts of domain ontology.
Procedure:

```
Do While i <= N
{
  Remove HTML tags /*Special characters from D_i.*/
  Remove less meaningful terms or stop words.
  D_i = D_i−S
  Apply Stemming and find the root words.
  w=w or prefix+w or w+suffix or Plural(w) or Tenseform(w)
  D_i = (w_i1, w_i2, ..., w_im)
  For j = 1 to m
  {
    Calculate tf (W_ij) = Count(w_ij, D_i)
    tagf (w_ij) = Present(w_ij, URL) + Count(w_ij, Tags)
    Get ontology entries C_jk containing W_ij
    Do while k <= C /*No. of concepts for W_ij*/
    {
      tf (C_jk) = Count(C_jk, D_i)
      tagf(C_jk) = Present(C_jk, URL) + Count(C_jk, Tags)
      Save W_ij and its set C_jk for D_i along with their scores in
   the Database Table.
    }
  }
}
  Repeat until i <= N
  {
    Repeat until j <= m
    {
      If w_j presents in D_i
      df (w_j) = df (w_j) + 1
    }
  }
    Do while j <= m
  {
    idf (w_j) = logN/df (w_j)
    Store df and idf for w_j in the Database Table.
  }
```

The collection of web documents is preprocessed by removing stop words and performs stemming. This results in a set of pure words for each document. For each word the term frequency is calculated in the content, in the URL address and also in specific tags such as Head, Title, Link, Anchor, etc. Then each word is mapped with the concepts of ontology. For each concept words the same concept frequency is calculated in the content, in the URL and also in the important tags. These values are indexed in the database table for the purpose of retrieval. In the last part of the algorithm the document frequency for each word and its inverse document frequency are calculated.

Algorithm: Semantic Query Conversion and Ranked Retrieval
Input: Query, Knowledge Base, threshold t
Output: Semantic Query, Retrieval of ordered relevant results
Parameters: Q-User Query, S-Stop words, QW-Words in the query,
C-Concepts of domain ontology, *NQ*-Number of words in the query
R-Number of web documents in the result set
tf-Term Frequency, *df*-Document Frequency,
idf-Inverse Document Frequency, *tagf*-Tag frequency
t-threshold, *Score*1-Term based score,
*Score*2-Tag based score
Procedure:

```
User Query Q = (Qw_1, Qw_2, ..., Qw_n)
Q = Q- {Special Characters}
Q = Q–S/*Remove Stopwords.*/
Q = Q or Stem(Q) /*Apply Stemming.*/
Do while i <= c /*Set of concepts in the domain ontology */
{
  If Concept(Q)=ci of Ontology
  Q = Q U ci
  Else if Synonym(Q) = S of Thesaurus
  Q = Q U Stem (Synonyms (Q))
  Else if Q = Link (D)
  Q = Q U Link (D)
}
Do while i <= NQ /*No. of words in the query*/
{
For j = 1 to N
{
If Qwi present in D_j
Insert D_j in the hash table for Qwi
}
}
Result={} RankResult={}
For i = 2 to NQ
/* No. of entries in the hashtable = No. of Query Words*/
{
Result=ResultU{hashtable(Q_w1)∧hashtable(Q_w2)∧..∧
 hashtable(Q_wi)}
}
For x = 1 to R /*No. of web documents in Result*/
For i = 1 to NQ
{
Score1 = Score1 + tf (Q_wi) * idf (Q_wi) * idf (Q_wi)
Score2 = Score2 + tagf (Q_wi)
}
Save Result, Q, Score1 and Score2 in a Database Table
If Result.Score1 > t
RankResult = RankResult U Result
}
Display the results from Rankresult in the order of score1
 followed by score2.
```

The user query is preprocessed here with stop words removal and stemming process. The query is expanded using the ontology concepts or with the synonyms of the query words or using the links of the web documents. Next the query expanded here is matched with the semantic web documents using the hash table structure. It will retrieve the intersecting set of the query matched results and concepts matched results. For the retrieved results, the similarity between the query and the result is calculated using Eq. (4). The retrieved results are ranked based on their similarity values and controlled by the threshold value t.

5. Implementation and experimentation

The proposed architecture of SBIRS is implemented using C#.net as Web- based system in visual studio 2010, NET framework 4, and SQL Server 2008 and executed on a Windows 7, 64 bit system environment with a Intel Core i5–2410 M CPU@ 2.30 GHz with 4 GB RAM and 500 GB hard disk. The test data are the web-documents collected from different domains like Food, Education, News and Health-care.

Processes of this experiment begin with the semantic extraction module to extract important concepts of the domains and transformed into semantic patterns. Next a total of 11 queries from four domains of the experiment data were fed into the SBIRS to perform semantic query extension. The searching is carried out with the semantic query and the semantic content. The results are compared with the keyword-based IR system-KBIRS (Unranked and Vector space ranked) to analyze the difference between them.

The various parameters used for analysis of this system are listed out in the Table 1.

The effectiveness of the information retrieval system is usually measured by the ratios Precision, Recall, F-measure and Average Precision. These values are calculated for each query then calculation of the Mean Average Precision (MAP) is derived. (David Grossman, 2009) .

Precision P is the ability to retrieve top-ranked documents that are mostly relevant.

$$\text{Precision} = \text{No. of relevant documents retrieved} / \text{Total No. of documents retrieved} \quad (5)$$

$$\text{Precison} = (\text{relevant documents} \cap \text{retrieved documents}) / \text{retrieved documents} \quad (6)$$

Recall R is the ability of the search to find all of the relevant items in the corpus. It is the ratio of the number of relevant documents retrieved to the total number of documents in the collection that are believed to be relevant.

$$\text{Recall} = \text{No. of relevant documents retrieved} / \text{Total No. of documents relevant} \quad (7)$$

$$\text{Recall} = (\text{relevant documents} \cap \text{retrieved documents}) / \text{relevant documents} \quad (8)$$

F-measure is the Harmonic mean of recall and precision.

$$F = 2PR/(P+R) = 2/(1/R + 1/P) \quad (9)$$

A set of queries are prepared manually and tested in both KBIRS and SBIRS with the help of precision and recall. The sample values are given in Table 2.

Table 1
Parameters used in the analysis.

Parameter	Meaning
M	Number of domains
N	Number of web documents
D	Web document
W	Words of web documents D
C	Concepts of domain ontology
QW	Query words
tf	Term frequency
tagf	Tag frequency
df	Document frequency
idf	Inverse document frequency
wt	Weight of the term
sc	Similarity coefficient
P	Precision
R	Recall
F	F-measure

Table 2
Precision and Recall values for 11 Queries of different domains.

Search Item	KBIRS		SBIRS	
	Precision	Recall	Precision	Recall
1	1	0.5	0.93	0.6
2	0.33	0.5	0.83	0.75
3	0.48	0.67	1	0.75
4	0.5	0.58	0.87	0.67
5	0.37	0.63	1	0.63
6	0.27	0.5	1	0.67
7	0.44	0.63	0.92	0.67
8	0.44	0.67	1	0.67
9	0.44	0.63	1	0.5
10	0.23	0.6	1	0.63
11	0.46	0.6	1	1

Fig. 5(a) and (b) show the difference between the average Precision and recall values of 11 Queries, Mean average precision and Mean Average Recall values for KBIRS, Vector-space KBIRS and the SBIRS respectively. The graph depicts that the SBIRS shows high precision and Recall compared to KBIRS and vector space KBIRS. From this graph it is inferred that the proposed SBIRS outperforms the KBIRS in terms of precision and recall.

The comparison of keyword based and Semantic based system in terms of Precision vs. Recall is given in Fig. 6. This graph shows the general inverse relationship between precision and recall remains for both systems and the Semantic based system is with high recall and precision.

The efficiency of the proposed Semantic based system is compared with Keyword based system with the values of Precision, Recall and F-measure. From the graph (Fig. 7) it infers that the proposed new system outperforms in all the above said measures than the keyword based.

The response time for different queries to retrieve the result set in SBIRS and KBIRS is compared in Fig. 8. This graph depicts that the time taken is high and varying rapidly for Keyword based where as it gradually varies with slight variation in Semantic based system.

5.1. Conclusions and future directions

The main goal of this research is to present a generic architecture to bridge the gap between IR and SW in the understanding and realization of semantic search. The idea of semantic search understood as searching by meanings and concepts solve the limitations of keyword based model.

The framework presented here accepts the natural language query which is then converted into semantic query by means of automatic link extraction algorithm, ontology and the external thesaurus. This query is then matched with the semantically annotated web database by means of an improved semantic indexing technique. Finally the retrieved results are presented to the user arranged by the order of their semantic relevance using a new dynamic ranking algorithm.

The strengths of SBIRS are investigated as Effective Semantic Query Conversion, Automatic Link Extraction Algorithm, Content Annotation Algorithm, Dynamic Ontology Creation, Semantic Hash Indexing Structure, Novel Ranking Strategy, Natural Language Query, Improved Precision and Recall, Generic Architecture and not domain specific.

Because of the decentralized and heterogeneous web, even on the same domain, it seems impossible for all web pages to use the same ontology. So study in semantic communication between ontology will be needed. As a side effect of an ontology-based approach, the problem of knowledge incompleteness is investigated.

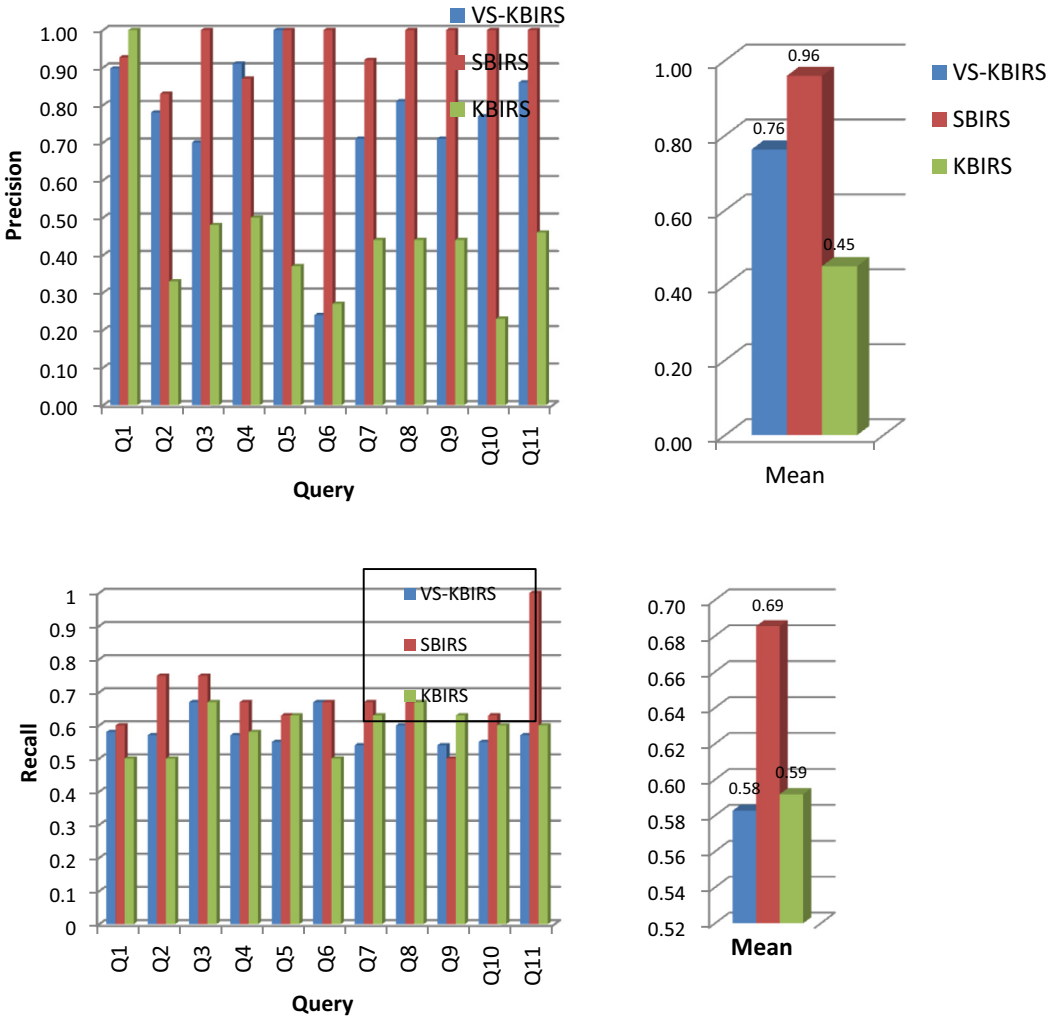

Fig. 5. (a) Result of Query matching-Precision Rate and (b) Result of Query Matching-Recall Rate.

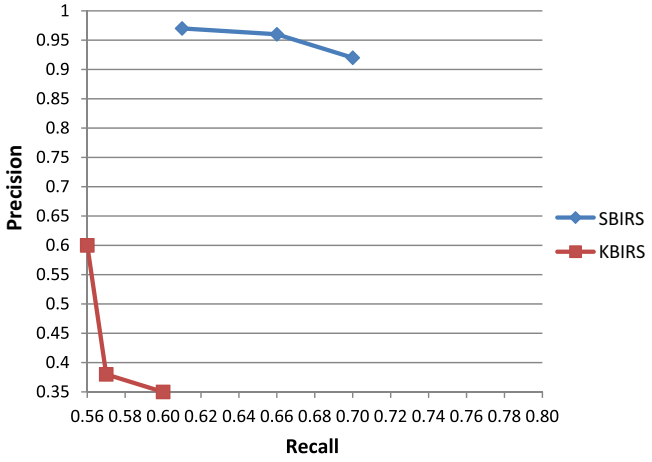

Fig. 6. Precision vs. Recall.

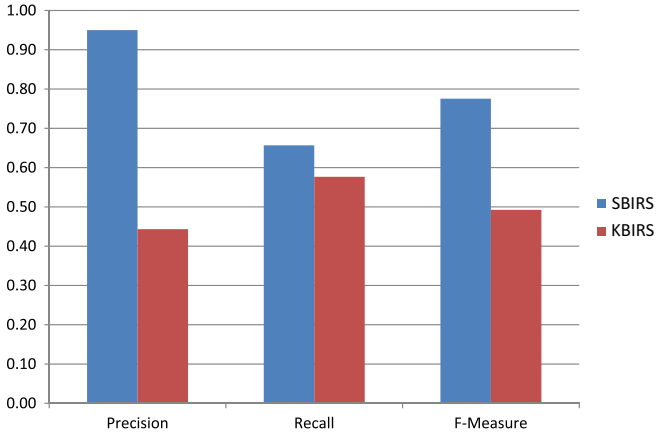

Fig. 7. Precision, Recall and F-measure.

6. Future directions

The current implementation can be extended and improved in the following areas.

- Initially the knowledge base can be encircled to support multiple languages.
- The extension of user preferences and item features through ontology properties enable the detection of further co-occurrences of interests between users and finds new interests, available for recommendations.
- As a further extension of this research, the practical feasibility is investigated for applying semantic search models to the cloud environment.

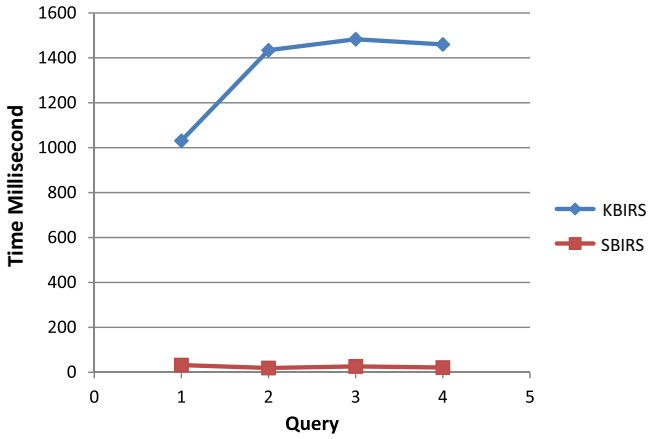

Fig. 8. Response time for different queries.

- This framework can be adopted for images as semantic imaging.
- This implementation can be further used in Cloud Environment for web services.

References

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Science American*, 29–37.

Castells, Pablo, Fern'andez, Miriam, & Vallet, David (2007). An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering, 19*(2), 261–272.

Chen, Ming-Yen, Chu, Hui-Chuan, & Chen, Yuh-Min (2010). Developing a semantic-enable information retrieval mechanism. *Expert Systems with Applications, 37*, 322–340.

Chu, Hui-Chuan, Chen, Ming-Yen, & Chen, Yuh-Min (2009). A Semantic-based approach to content abstraction and annotation for content management. *Expert Systems with Applications, 36*, 2360–2376.

Croft, W. B. (1986), User-specified domain knowledge for document retrieval. In: *Proceedings of the 9th annual international ACM conference on research and development in information retrieval (SIGIR 1986)* (pp. 201–206). Pisa, Italy.

Fern'andez, Miriam, Cantador, Iv'an, L'opez, Vanesa, Vallet, David, Castells, Pablo, & Motta, Enrico (2011). Semantically enhanced information retrieval: An ontolgy-based approach. *Web Semantics: Science, Services and Agents on the World Wide Web, 9*, 434–452.

Jo rvelin, K., Kekalainen, J., & Niemi, T. (2001). Expansion tool: Concept-based query expansion and construction. *Information Retrieval, 4*(3–4), 231–255.

Köohler, Jacob, Philippi, Stephen, Specht, Michael, & Rüuegg, Alexander (2006). Ontology based text indexing and querying for the semantic web. *Knowledge Based Systems* (19, pp. 744–754). Elsevier, Science Direct.

Lamberti, Fabrizio, Sanna, Andrea, & Demartini, Claudio (2009). A relation-based page rank algorithm for semantic web search engines. *IEEE Transactions on Knowledge and Data Engineering, 21*(1), 123–135.

Lei, Yuangui, Uren, Victoria, & Motta, Enrico (2006). SemSearch: A search engine for the semantic web. In *EKAW 2006* (pp. 238–245). Springer.

Li dong Finin, L., Joshi, T. W., Pan, A., Scott Cost, R., Peng, R., et al. (2004). Swoogle: A search and metadata engine for the semantic web. *CIKM 2004*, 652–659.

Li, Yufei, Wang, Yuan, & Huang, Xiaotao (2007). A relation – Based search engine in semantic web. *IEEE Transactions on Knowledge and Data Engineering, 19*(2), 273–282.

Ochoa, Xavier, & Duvai, Erik (2008). Relevance ranking metrics for learning objects. *IEEE Transactions on Learning Technologies, 1*(1).

Ophir Froeder, A. (2009). *Information Retrieval - Algorithms and Heuristics*. Springer. International Edition.

Rudi Cilibrasi, I., & Paul Vitan'nyi, M. B. (2007). The Google similarity distance. *IEEE Transactions on Knowledge and Data Engineering, 19*(3), 370–383.

Salton, G., & McGill, M. (1983). *Introduction to modern information retrieval*. McGraw-Hill.

Singh, Ramesh, Dhingra, Dhruv, & Arora, Aman (2010). SSCHISMA web search engine using semantic taxonomy. *IEEE Potentials*, 36–40.

Yi, Jin., Zhuying, Lin., Hongwei, Lin. (2008), The research of search engine based on semantic web. In: *International symposium on intelligent information technology application workshops*.