Ontology based Semantic Indexing Approach for Information Retrieval System

Sajendra Kumar,
Department of Computer Science,
IIMT IET, Meerut, India

Ram Kumar Rana,
Department of Computer Science,
IIMT IET, Meerut, India

Pawan Singh,
Department of Computer Science,
IIMT IET, Meerut, India

ABSTRACT

This paper shows how the gap between the texts based web pages and the Resource Descriptive Framework based pages of the semantic web can be bridged by ontologies. Most traditional search engines use indexes that are engineered at the syntactical level and come back hits based mostly on straightforward string comparisons or use the static keyword based indexing. However, the indexes don't contain synonyms, cannot differentiate between homonyms ('mouse' as a Pointing device vs. 'Mouse' as a living animal) and users receive completely different search results after they use different conjugation varieties of identical word. During this work, we have a tendency to gift a system that uses ontologies and Natural Language Processing techniques to construct index, and therefore supports word sense disambiguation. Therefore the retrieval of document that contains equivalent term as the context demands is achieved to provide efficient search engines through ontological indexing.

General Terms

Semantic indexing theory, Architecture, Algorithm.

Keywords

IR Indexing, Semantic index, Ontology, Semantic search.

1. INTRODUCTION

Nowadays the growing importance of Internet and the World Wide Web has made indexing as one of the most important research fields. Many different index structures, compression techniques and retrieval algorithms have been proposed in the last few years. More importantly, these proposals have been widely used in the implementation of document databases, digital libraries, and web search engines. In any Information Retrieval technique, the user issue a query that contains terms and every documents containing terms as per the query are retrieved. Though all the documents are often sequentially scanned to search out terms within the user query, this system is extremely inefficient for giant document collections. So, all Information Retrieval techniques use some sort of indexing to hurry up the search. In full text indexing, virtually each word in the document is employed as an index term.

Indexing [1] can be defined as a process that collect, parse and store data to facilitate fast and accurate information retrieval. Typically, information is retrieved by matching terms in documents with those of a user query. However, a lexical matching strategy is inadequate. Since there are typically many ways to specify a given concept (synonymy), the literal terms in a user's query might not match those of a relevant document. Additionally, most words have multiple meanings (polysemy), therefore terms in a user's query can literally match terms in irrelevant documents. An improved approach would permit users to retrieve information on the basis of a conceptual meaning.

Semantic Indexing tries to beat the problems of lexical matching by using statistically derived conceptual indices instead of individual words for retrieval. Semantic Indexing assumes that there's some underlying or context structure in word usage that's partially obscured by variability in word selection. The foremost well-liked full text indexing structure is inverted index [1,2] and multi-dimensional indexing Spatial access method (SAM) and Point access method(PAM) technique. The opposite indexing technique that is widely used and that isn't a full text indexing technique is Latent Semantic indexing (LSI) [3].

1.1 Inverted Indexing

Inverted indexes associate each word in the text with a list of pointers to the positions where the word appears in the document. For each term, the list of all documents which contained that term and additional information about that term like the frequency of the word are stored. The terms are organized as a B+ tree data structure for quick lookup. When an enquiry is performed, the B+ tree storing the terms is queried with every term within the user question. The scales of the inverted index are often huge for giant document collections. There are many issues to be addressed on compressing of inverted index. Since most of the database within the inverted index is utilized for storing document IDs and offsets. An index compression technique is often used to restrict the scale of the inverted index.

Figure1. shows an inverted index construction, considering all words together with stopwords. When querying, the lists are extracted from the inverted index and then merged.

Vocabulary grows sub-linearly with the gathering size; the list of occurrences will be terribly massive. The whole inverted index will take a major fraction of the house occupied by the particular collection. An inverted index doesn't slot in main memory for an online assortment, thus many partial indices are designed. Every partial index represents solely a subset of the gathering and is later merged into the complete inverted index.

1.2 Multidimensional Indexing

Multidimensional indexing structures are information structures that support indexing and retrieval of objects that have quite dimension. Multidimensional information embodies points, line segments, rectangles, and polygons in second, 3D or higher. Storage and retrieval of multidimensional information is very important in several businesses, scientific and engineering application.

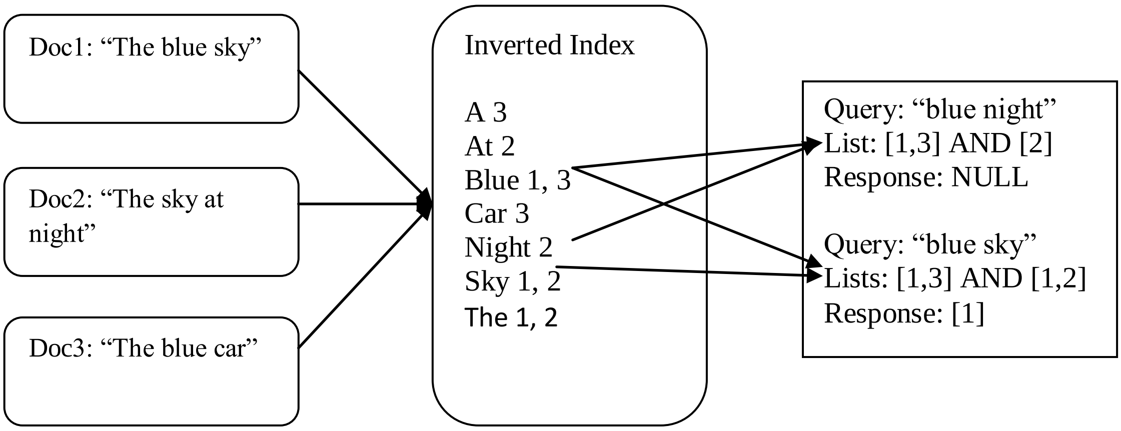


Figure1. Construction of an Inverted Index

Multidimensional access strategies are often classified into Point Access strategies (PAM) and Spatial Access Methods (SAM). PAM is primarily designed to index and search multidimensional points that don't have any spatial extension. SAM is intended for objects that have spatial extent like lines, polygons or higher dimensional polyhedron. Some standard PAM embody Grid files, Quad-trees [4] and k dimension trees. The grid file could be a multidimensional array used as an index to things that have multiple dimensions. This method relies on hashing and guarantees that any records are often retrieved by at most two disk accesses. This is often done by creating use of a grid directory consisting of grid blocks and every one records in a very block are stored within the same bucket. The grid itself is maintained within the main memory and is represented as one-dimensional arrays referred to as scales. The quad tree is recursively defined: split the current information into four quadrant and recursively construct quad-trees for every quadrant. In k dimension trees, at every intermediate node, the k-dimensional area is split into 2 elements by a (k-1) dimensional hyper plane. The direction of the split alternates between the k potentialities from one tree level to future. The foremost standard SAM techniques embody R trees and R+ trees. In R-trees, a group of hierarchically nested Minimum Bounding Rectangles (MBR) is maintained. Every node of the R-tree stores a variable range of components. Every component stores the simplest way of identifying the kid node and a MBR of all the weather within the kid node. R+ trees are an extension of the R-trees, whereby overlapping of internal node is avoided by inserting an object into multiple nodes if necessary.

1.3 Spatial Indexing

Spatial indexing are those indexing methods that exploit geo-tagging procedures to categorise documents with respect to geographic space. A spatial index is a special access method used to retrieve data from within the data-store. Spatial indexes allow users to treat data within a data-store as existing within a two dimensional context. A spatial index is a grid divided into a number of rectangles or cells. All cells have the same width and height. All cells, together, map a single large rectangle. Two fields are required from any record to map the record to the grid effectively the (X, Y) coordinate pair. These types of indexing methods are used in spatial search engine. A Spatial Search Engine on the other hand works like this. First the data to be searched is aggregated or compiled. This data is then *spatially enabled*. For example, your home location can be defined spatially by its latitude and longitude coordinate. Or in three dimensions by modelling that object by its location and its elevation. Hundreds of millions of spatially-enabled objects can then be spatially indexed such that fast geometric search operations can be applied.

All the above discussed indexing techniques [5] were based only on terms extracted from the document that's why these were not smart enough to build a semantic index. Now we discuss few more efforts in the direction of smart indexing technique given below.

1.4 Latent Semantic Indexing

In most IR techniques, the terms within the documents and therefore the question terms are actually matched. Such techniques fail to acknowledge synonymy (multiple words having identical meaning) and polysemy (words having multiple meanings). This drawback is addressed in Latent Semantic Indexing (LSI). In LSI, an occasional rank Singular Worth Decomposition (SWD) of the document-term matrix is obtained. The projection into the latent semantic house is chosen such that the representations within the original house are modified as very little as attainable when measured by the sum of the squares of the variations.

2. BACKGROUND

In this paper, a summary of the previous research work for the indexing is given. In indexing many techniques have been proposed already, but these techniques were less efficient in the term of producing quality results.

In paper [11], work proposed by the author was clustering algorithm on the bases of threshold. In this algorithm documents choose the cluster on the bases of threshold value define in the clustering algorithm. If two or more document has same threshold value then they classified in the same cluster. But if the threshold value of the document is less than the defined value then they choose the different cluster. If the threshold value of the document is high then they join the one cluster only. The problem of this algorithm was how to define the threshold.

Another paper [12], the author proposed a new indexing technique known as double indexing. This technique was for search engine based on campus net (CNSE). Campus net has crawl machine, Chinese automatic segmentation, and index and search engine. In this technique indexing is done by word index, after the word indexing documents do the order clustering. During the searching search engine first extract the doc-ID from the word index the go to its position in word index. in this technique searching of document is time consuming because indexing is done in two phase.

3. PROPOSED WORK

The model proposed in figure 2 makes use of the prevailing resources like lexical database WordNet [6] for English language, semantically annotated documents of various domains.

The search model provides ontology based indexing with two modules; it takes the web page from the web which is downloaded by the multithreaded downloader. To understand the various contexts of the terms fetched from these web pages we use of WordNet database of English vocabulary. Once the contexts are finalized the term is matched with ontology of related class. After understanding term definition it proceeds to link ideas, with the term. One can call this finding the synonyms or normalizing the term. This context is stored in data structure of index with context column. That will help in finding quality results for search engines. The modules of our architecture are discussed below.

3.1 Multi-threaded Downloader

A crawler [7] or bot or spider is a program follow URLs and traverse the web in order to download web pages through multi-threaded downloader. A multi-thread crawler generate more than one thread at a simultaneously those request web pages through http request. In order to fetch a Web page, we need an HTTP client which sends an HTTP request for a page and reads the response. The main aim of a crawler or downloader is to populate the web page repository.

3.2 Web Page Repository

It is a collection of heterogeneous pages submitted by web crawler means web repository consist of web pages in different formats like HTML, XML, PHP, ASP and RSS etc. These documents are main resource for index module.

3.3 Indexing Module

During the process of **indexing** documents are prepared to be used by an Information Retrieval. This means preparing the document collection into an easily accessible representation of documents. The transformation from a document text into a *representation of text* is known as **indexing** the documents.

Transforming any document into ontology based index involve many steps to be performed. These steps are shown with the help of few sub modules discussed below:

3.3.1 Parse Page

It takes web page from repository and parse that in order to extract keyword and terms. Once a page has been fetched, we need to parse its content to extract information that will feed and possibly guide the future path of the indexing. Parsing may imply simple terms extraction or it may involve the more complex process of tidying up the HTML content in order to analyze the HTML tag for context and meaning. Parsing might also involve steps to convert the extracted keyword to canonical form, remove stopwords from the page's content and stem the remaining words. The parsing of the text can be very basic to very advanced using different Natural Language Processing techniques. We will give a listing of the most interesting techniques:

3.3.1.1 Stop listing

Every page contains a large number of stopwords, such as"it","can","the", etc. Stop listing is the process of removing those stopwords from the text.

3.3.1.2 Stemming

The stemming process normalizes words by conflating a number of morphologically similar words to a single root form or stem. For example," connect", "connected" and" connection" are all reduced to "connect". Implementations of the commonly used Porter stemming algorithm are easily available in many programming languages.

3.3.2 Define Context

After parsing the web page the indexer module tries to define the context [8] of term. It consults ontology and word net module to give a final definition to any term. It checks synonyms of the term to justify what context it has been used? Once the meaning is clear then it matches same term with its possible ontological class. Finally this module shape index of that term and submit to next module.

3.3.3 Ontologies

One can find many different definitions for the concept of ontology applied to information systems, each emphasizing a specific aspect its author judged as being more important. For instance, Gruber [9] defines an ontology as a "*formal specification of a conceptualization*" or, in other words, *a declarative representation of knowledge relevant to a particular domain*. Uschold and Gruninger [10] define ontology as a shared understanding of some domain of interest.

3.3.4 WordNet

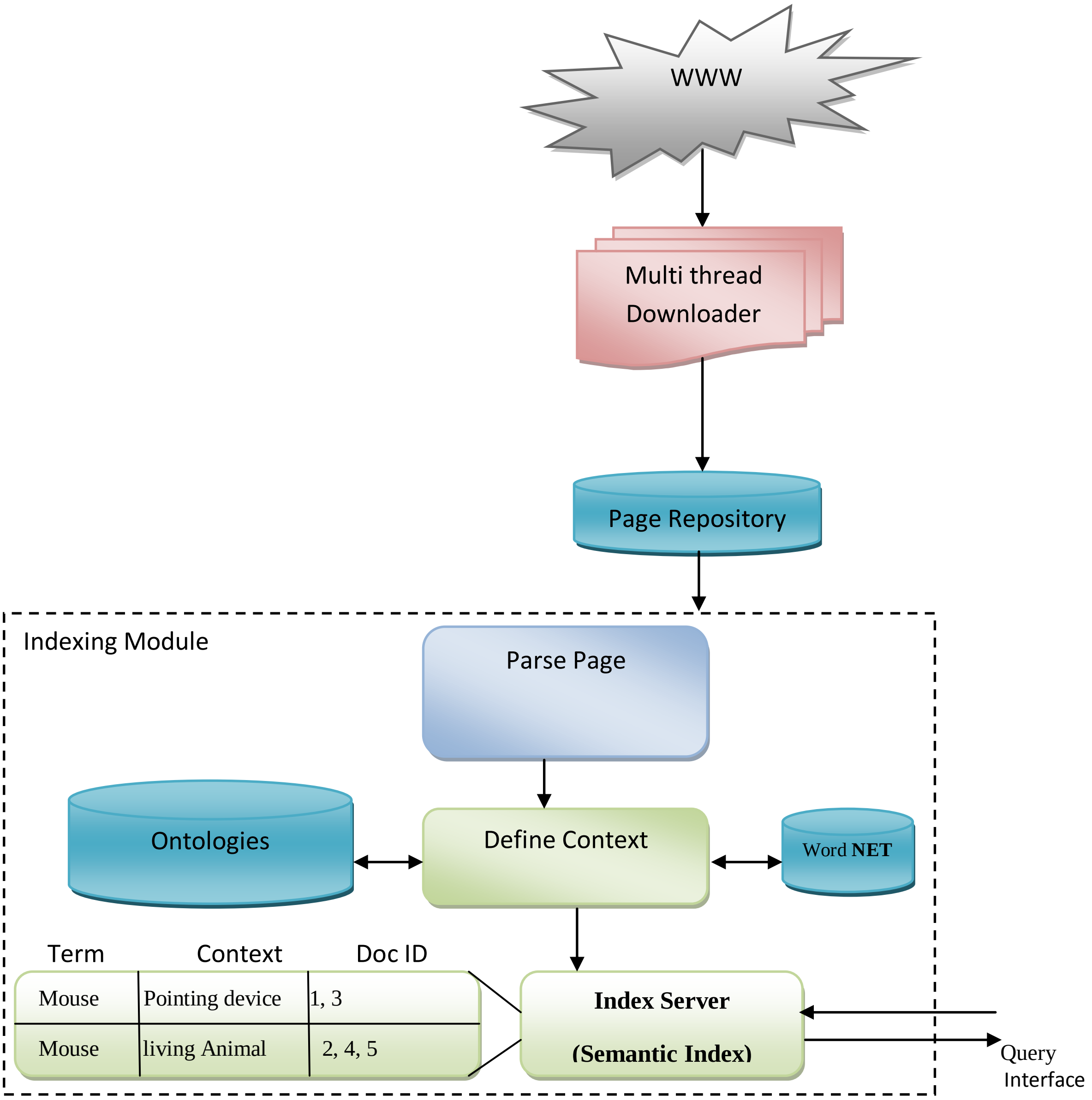
Wordnet[6] is an electronic lexical database for English, where various senses (possible meanings of a word or expression) of words are put together in a set of synonyms. Such a set of synonyms is called a Synset and represents the notion of a concept in WordNet. A Synsetis represented by a line in a WordNet position data file. A Synset contains a set of Words, each of which has a sense that names that concept (and each of which is therefore synonymous with the other words in the Synset).

3.3.5 Index Server

This is most important module or may be said sole output of indexing. In this module the final semantic index is stored. This index is different than simple text index because its take care of context and meaning of term. There are many data structure like Hash table and different types of trees like R-tree, B-tree to represent this. An example is shown how this index store terms with their context for term *mouse* [7].

3.4 Query Interface

This module provide interface between index and search engine user. It enables a user to submit input in the form of query and produce output in the form of displayed result. An index based on above discussed architecture help ontological query processing [13].



Term	Context	Doc ID
Mouse	Pointing device	1, 3
Mouse	living Animal	2, 4, 5

Figure 2: Proposed Architecture for Semantic Indexing based on ontology

4. ANALYSIS OF PROPOSED AND THE EXISTING INDEXING

The problem with the traditional indexing technique is that how to organise the keyword or term for indexing. If the term or keyword is not well defined then the searching of document is time consuming and the returning result may not be related to the user requirement. These days, there is a lot of growing interest in *ontologies* for managing data in information retrieval systems. In fact, ontologies provide semantic for understanding the meaning of data. They are broadly used in all information retrieval systems to overcome the problems caused by the term or keyword based indexing. In our technique as compare to the traditional indexing technique (inverted index) result of the query is much effective and meaning full or in context.

5. CONCLUSION

To construct ontology based Semantic index is an important issue in web search engines. In this paper we showed how to build semantic index for web search engine. We have discussed about the proposed architecture for ontology based semantic indexing. We have also discussed about various research findings on how to efficiently build index by using different techniques i.e. inverted index, spatial access method and Latent Semantic indexing.

We have described a framework for design semantic indexing for information extraction and retrieval that aims to find effective and context based information from unstructured and millions of documents. This research has only begun to explore the possibilities of semantics-based indexing for deriving structured and meaningful information. A rough estimate of support values for the existing and the proposed system clearly depicts the better performance of the existing system.

6. REFERENCES

[1] Clarck C., Cormack, G: Dynamic Inverted Index for a Distributed Full text Retrieval System. Tech. Rep MT-95-01, University of Waterloo, Feb-1995.

[2] Ajit Kumar Mahapatra, Sitanath Biswas,"Inverted Index: Types and techniques", International journal of Computer science Issues, Volume-8,Issue-4, No.1, July 2011.

[3] Ding, C., A Similarity-based Probability Model for Latent Semantic Indexing, Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, pp. 59–65.

[4] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf (2000). Computational Geometry (2nd revised ed.). Springer-Verlag. ISBN 3-540-65620-0. Chapter 14: Quadtrees: pp. 291–306

[5] Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. Information Retrieval: Implementing and Evaluating Search Engines. MIT Press, Cambridge, Mass., 2010.

[6] WordNet-Online dictionary and hierarchical thesaurus Obtained through the Internet http://www.wordnetonline.com [accessed 28/12/2009]

[7] Ram Kumar Rana, Nidhi Tyagi, "A Novel Architecture of Ontology-based Semantic Web Crawler", International Journal of Computer Applications (0975 – 8887) Volume 44– No18, April 2012.

[8] N. Chauhan, A. K. Sharma, "Context Driven Focused Crawling: A New Approach to Domain-specific Web Retrieval", paper presented at International Conference on information & Communication Technology (IICT), July, 2007.Dehradun.

[9] Thomas R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (1993), no. 2, 199–220.

[10] M. Ushold and M. Gruninger, Ontologies: Principles, methods and applications, The Knowledge Engineering Review, 1996.

[11] Oren Zamir and Oren Etzioni. Web Document Clustering: Afeasibility demonstration. In the proceedings of SIGIR, 1998.

[12] Changshang Zhou, Wei Ding, Na Yang, Double Indexing Mechanism of Search Engine based on Campus Net, Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06).

[13] Sajendra Kumar, Ram Kumar Rana, Pawan Singh, "A Semantic Query Transformation Approach Based on Ontology for Search Engine", International Journal on Computer Science and Engineering (IJCSE), May 2012. (688-693).