Scalable Semantic Search with Hybrid Concept Index over Structure Peer-to-Peer Network

wenhui Ma, gang Wang, jing Liu
College of Information Technology and Science
Nankai University
{wenhuima_nk}@hotmail.com

Abstract

The primary challenge in developing a peer-to-peer(P2P) file sharing system is implementing an efficient keyword search mechanism. Current keyword search approaches for structured P2P networks are built on the distributed inverted index by keywords. However, when executing multiple-attribute queries, they suffer from the problem of unscalable bandwidth consumption. Moreover, these approaches only support literally word match, not taking into account the meaning of word. In this paper, we propose an efficient keyword search mechanism over structure P2P network. Peers use a shared ontology to describe the content of a document and the subject of a query. A distributed hybrid concept index is constructed, which efficiently supports the query routing and matching, and avoids the intersection of inverted list among peers, which is cause of unscallabe network bandwidth consumption. Based on the semantic similarity between the subjects of queries and the contents of documents, peers can get results matching their queries semantically, instead of literally word match. Simulation experiments show that keyword search with the approach proposed in this paper is much less on bandwidth costs and much higher on retrieval perform than that based on standard inverted index by keywords.

1. Introduction

Recently, the P2P networks have gained tremendous interest for information resource**,** such as documents, video, audio, and image, sharing over Internet. Although the P2P infrastructure provides a scalable alternative to conventional central server based approaches, implementing efficient information retrieval in such large-scale P2P network remains challenging problem. Structure P2P networks, such as Chord[1] and CAN[2], use distributed hash table (DHT) addressing some of the scalability and reliability problems that exist in earlier P2P networks such as Naspter[3] and GNUtella[4]. They impose constraints both on the node graph and on resource placement to enable discovery, and can lookup an existing information resource in a small bounded number of hops (O(logN)) for a network consisting of N nodes. But, structure P2P networks only offer a simple interface for storing and retrieval (key, value) pairs, and hence not suitable for keyword search. While, as they actually implement DHT over them, keyword search can easily be implemented by constructing distributed inverted index by keywords. This approach is adopted by some proposals[5] [6] to implement the full-text searching functionality in the structure P2P networks**.** However, search based on keyword index over structure P2P network suffers from the storage constraints when there is large number of documents in the network, and unscaleable bandwidth consumption problem when executing multi-keyword search[7].

Moreover, another challenging problem with keyword index is that it does not take into account meaning of word, and search is only literally matching words (or words combination) in documents with those present in a user's query. This can lead to poor retrieval performance due to ambiguity of natural language in two facts. First, because many words have multiple meanings, many unrelated documents may be retrieved just because they matched some of the query keywords. Second, because the same concept can be described by multiple words, relevant documents that do not contain any of the query keywords will not be retrieved.

In this paper, we propose an efficient keyword search mechanism over structure P2P network, We introduce a generic ontology, WordNet[8], which is shared by each peer. Through ontology mapping and sense disambiguation for words, peers generate a set of concepts extracted from the ontology to describe the content of a document shared and the subject of a

query. So, some of the problems caused by ambiguity of nature language can be addressed. A distributed hybrid concept index is constructed, which combines the global concept index with local document index. It avoids the intersection of inverted list among peers, which is cause of unscalable network bandwidth consumption, and efficiently supports the query routing and matching. Query with multiple keywords is executed by the process of concept matching that is based on the semantic similarity between the subject of a query and the contents of documents, which tackles some problems posed by the riches of natural language and improves the retrieval performance of query.

The remainder of this paper is organized as follows. Related work is discussed in Section 2. Section 3 describes ontology, and ontology based hybrid concept index building is discussed in section 4. Section 5 describes keyword search based on concept match. The simulation experimental results are presented in section 6. Section 7 concludes this paper.

2. Related work

Some solutions have been proposed to overcome the unscalable bandwidth consumption for keyword search in DHT based P2P system. [9] proposes a full-text retrieval engine, ALVIS PEERS, it can scale to a very large number of peers. ALVIS PEERS limits the generated traffic when processing queries through reducing the size of posting list associated with indexing component. To obtain short posting lists, ALVIS PEERS indexes sets of terms that occur simultaneously in documents from the collection vocabulary. Through identifying discriminative term combinations at indexing time, ALVIS PEERS avoids performing long posting list intersections at query processing time that generates unscalable network traffic. [10] has reduced bandwidth consumption by pursuing a hybrid between partitioning by keywords and partitioning by documents, and implements keyword search using multi-level partitioning (MLP) in P2P system. However, MLP is designed and implemented on top of SkipNet[11], relying on a node group hierarchy, and it can not apply to other DHT based P2P system.

For the index scheme, most of the existing works in P2P information retrieval use keyword index based approaches[5] [6] to support quick query execution, especially for short queries. However, the keyword index alone can only support simple retrieval tasks and would be hard to support sophisticated retrieval algorithms. In contrast to keyword index, document index can support many complex retrieval tasks because the information about the whole document is always available together. There also has been hybrid index scheme[12], which combines the keyword index with document index. However, all these index schemes are only literally matching terms in documents with those present in a user's query, they can not deal with any semantic information in document and query. In [13], Tang proposed PeerSearch, which only needs to search a small number of nodes to identify matching documents through combination of index placement and query routing. It is built on top of the CAN and leverages the Latent Semantic Index (LSI) to capture the semantic relation between terms. PeerSearch represents documents and queries as vectors and measure the similarity between a query and a document as the cosine of the angle between their vector representations. PeerSearch stores a document index in CAN using its vector representation as the coordinates, so indices stored close to each other are also close in semantics. This unifies the problem of semantic-based search with routing in an overlay network.

3. Ontology

In computer science, the famous definition for ontology is Gruber's definition[14] "an ontology is an explicit specification of a conceptualisation". Therefore, an ontology defines a set of representational terms called concepts, as well as interrelationships among these concepts. Ontology aims at defining meaning of the terms and relations among these terms for a target domain, and providing a commonly understanding between users or applications for that domain.

WordNet is an on-line lexical reference system developed at Princeton University. WordNet attempts to model the lexical knowledge of a native speaker of English[15]. It provides a more effective combination of traditional lexicographic information and modern computing[8]. WordNet can also be seen as an ontology for nature language. It is consisted of synonym sets called synsets, and each synset represents a single distinct sense or concept. WordNet stores information about words that belong to four parts-of-speech: nouns, verbs, adjectives and adverbs. In WordNet 2.0, there are 152059 words organized in 115424 synsets, approximately 20% of the words in WordNet are polysemous; approximately 40% have one or more synonyms[8]. WordNet 2.0 features a rich set of 333612 relation links among words, between words and synsets, and among synsets.

Because noun bears more important semantic, in this paper we only use noun synsets of WordNet as a shared ontology by each peer of P2P network. The

major semantic relations for noun synsets defined in WordNet 2.0 and their statistics is listed in Table 1.

Table 1. Major semantic relations among nouns and statistics in WordNet 2.0

Semantic relation	Count
Hyponym/Hypernym (is-a/has-a)	93186
Substance Meronym/Holonym (substance of / has substance)	607
Part Meronym/Holonym (part of / has part)	7793
Member Meronym/Holonym (member of / has member)	12140

The relations connecting synsets are invertible, and the meanings of them are described as follows:

Hyponym/Hypernym: it represents the synset inclusion and can be expressed as " is a" or "is a kind of". If synset A is a kind of synset B, then A is the hyponym of B, and B is the hypernym of A.

Meronym/Holonym: this relation is used to represent the part-whole relation between synsets. The Meronym/Holonym relation is sub classified in three relations, substance of/has substance, part of/has part and member of/has member. If synset A is a part of (substance of or member of) synset B, then A is the meronym of B, and B is the holonym of A.

In this paper, we use the WordNet RDF/OWL representation[16] for WordNet 2.0 as global ontology shared among peers of P2P network for document indexing and query processing. The WordNet RDF/OWL representation is a W3C working draft, and it use RDF triples to represent the synsets and relations between them. This WordNet schema has three main classes: Synset, WordSense and Word. The first two classes have subclasses for the lexical groups present in WordNet, e.g. NounSynset and VerbWordSense. Each instance of Synset, WordSense and Word has its own URI. There is a pattern for the URI so that it is easy to determine from the URI the class to which the instance belongs. For example, The URI for an instance of NounSynset is:

"http://www.w3.org/2006/03/wn/wn20/instances/synset-computer-noun-1"

It represents this NounSynset containing a WordSense which is the first sense of the word "*computer*". In this paper, we only use the noun synset of WordNet RDF/OWL representation. Each noun synset is a type of *Rdfs:Class*. The semantic relations between noun synsets include hyponym, part meronym, substance meronym and member meronym, and each of them is a type of *rdf:Property*. More detailed information on WordNet RDF/OWL representation can refer to [16].

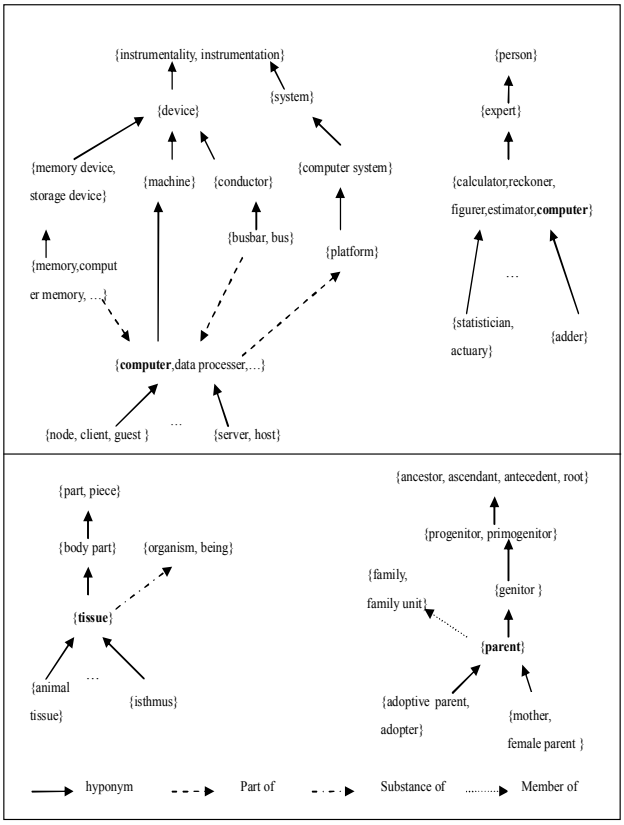

Figure 1. Extracted from WordNet illustrating the words and relations

4. Ontology based hybrid concept index building

Inverted index by keywords usually assumed that words in document are mutual independent and the meaning of words and semantic relations among them are not taken into account. In this paper, WordNet is introduced in document index constructing. Combining the concepts description for documents with inverted index by concepts, a hybrid concept index is constructed. It provides an effectively supporting for the scalable semantic search.

4.1. Concept Description for Document

Each peer generates a concept set from ontology to describe the content of a document shared to P2P system. Firstly, a set of words(and words combination) S_w is generated for the document, while "stop" words are eliminated, and the remaining words are stemmed so that there is only one grammatical form (or the stem common to all the forms) for a given word. Then, we map the keyword set S_w to the WordNet and extract the concepts containing these keywords.

Because of the ambiguity of natural language, a word maybe belongs to several concepts of WordNet. As shown in Figure 1, the word *computer* is appearing in two synsets that are represented by *{**computer**, computing machine, computing device data processer.…}* and *{calculator, reckoner, figurer, estimator, **computer**}* respectively. But the two synsets indicate two different concepts. So, the word sense disambiguation is needed. In this paper, we use the WordNet context of a word to determine which concept the word should belong to. The main idea is that the number of words in WordNet context of a word will determine appropriate senses for this word, and these words having direct semantic relations with this word in WordNet. Let $Context_w(c_i)$ represents the WordNet context of word w that is appeared in the concept c_i of WordNet.

Definition: *the WordNet context of the word w for the concept c_i is a set of words of WordNet,*

$Context_w(c_i)=\{synonym_w(c_i) \bigcup (synonym(meronym_w(c_i)) \bigcup synonym(holonym_w(c_i)) \bigcup synonym(hyponym_w(c_i)) \bigcup synonym(hypernym_w(c_i)) \}$

For example, from Figure 1 the WordNet context of the word *computer* for the concept *c' {calculator, reckoner, figurer, estimator, computer}* is:

$Context_{computer}(c')=\{$ *calculator, reckoner, figurer, estimator, expert, statistician, actuary, adder,…}*

For eliminating ambiguity of a word, we calculate the sense-score of this word. The sense-score is the number of words, which belong to a concept, appearing in the WordNet context of this word. The sense-score indicates how good a word belongs to a concept of WordNet in a document. Let $m(w,c_i)$ be the sense-score of the word w for concept c_i.

$$m(w,c_i) = \left| context_w(c_i) \bigcap S_w \right|$$

Here S_w is the word set of the document generated above. The sense-score determines the intended senses of a word and a corresponding concept is extracted from WordNet that interprets the meaning of this word.

In addition, there are another two cases that should be mentioned. One is that several words maybe have the same sense (synonym) and appear in the same concept of WordNet. We only use a concept representing these words. Another is that some words do not appear in any concept of WordNet, i.e. there are not concepts containing these words. This case usually occurs. The reason is that natural language is rich and quickly developing and WordNet can not cover all words of natural language. For this case, we transform these words as new concepts to describe the contents of documents, which is useful for query match later. Because the new concept does not belong to the ontology, it has no relations with any other concepts of WordNet.

According to sense-scores of words, peers generate a concept set S_c extracted from WordNet for each document shared, and all concepts of which have the maximum values of sense-score. The concept sets denote the intended senses of words and represents the contents of documents.

4.2. Hybrid Concept Index building

Based on the concept sets of the documents, we propose a distributed hybrid concept index. Each peer use concepts as the keys of DHT, hashing document metadata in the peers that are responsible for these keys. A global distributed inverted index by concepts is constructed for all documents over structure P2P network. Each peer stores inverted lists for some concepts. For each document d in the inverted list for a concept c, peer also stores the concept set of d locally. As a result, a distributed hybrid index is constructed, which includes two parts: one is global index that indicates which documents the concept appears in; the other is local index that indicates which concepts appeared in the document. Figure 2 shows the hybrid index structure.

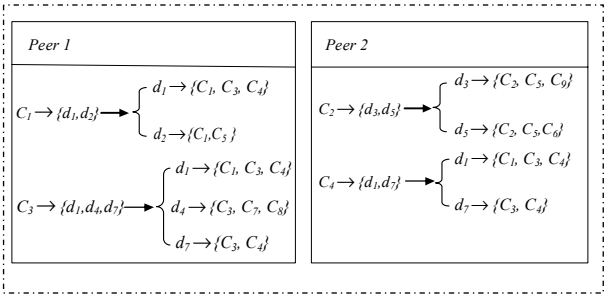

Figure 2. Hybrid concept index
($d_1,d_2,\ldots$ are documents and $C_1,C_2,\ldots$ are concepts)

The hybrid concept index structure can accurately locate the target documents through global index with DHT. And in the target peer, query can be matched locally without consulting other peers, which avoids the intersection of inverted lists among peers when executing multiple keywords search. Also hybrid concept index make foundation for semantic search over structure P2P network.

5. Keyword search based on concept match

In this paper, the query consists of multiple keywords. Queries are posed by peers and the operation to documents is also applied to queries. A concept set Q_c for a query is generated. Using each

concept of Q_c as the key of DHT, the query is sent to the peer p_i that is responsible for the key. As described in above section, in peer p_i the concept sets of the documents are also stored, so query is matched based on semantic similarity between the contents of documents and the subject of the query.

To be able to define the similarity of a document's content and a query's subject, which are both represented as a set of concepts, we first define the similarity measure between concepts. Li in [17] has compared different similarity measures and has proposed that for measuring the similarity between concepts in a large and generic semantic net, such as WordNet, semantic similarity considers to be determined by the shortest path length as well as the depth of the subsumer. He proved that the following similarity measure yields the best results:

$$sim(c_1, c_2) = \begin{cases} e^{-\alpha l} \cdot \dfrac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} & \text{otherwise} \\ 1 & \text{if } c_1 \equiv c_2 \end{cases} \quad (1)$$

Here c_1 and c_2 are concepts of WordNet; l is the length of the shortest path between c_1 and c_2. h is the level in the tree of the direct common subsumer from c_1 and $c2$. $\alpha > 0$ and $\beta > 0$ are parameters scaling the contribution of shortest path length l and depth h, respectively. In this paper, we only use the *hyponym/hypernym* relation of WordNet to measure the similarity of concepts. So all the concepts(synsets) are organized in a tree-like hierarchical structure. So, we use Formula (1) as our similarity function of concepts.

Before calculating the relatedness between a document and a query, we need weight each concept of S_c. This weight quantifies the important of the concept for describing the contents of document. In this paper, we extend the *tf*idf* scheme [18], which is usually used to compute the word weight in the Information Retrieval, to calculate the concept weight, and proposes the *cf*idf* scheme. The *cf* represents the appearance frequency of concept in a document, and the *idf*, which is invert document frequency of concept, represents how often the concept occurs in other documents in the system.

Let $freq_{ij}$ be the raw appearance frequency of keyword k_i, which belongs to the concept c, in document d_j. Then we can calculate the $freq_{cj}$, the appearance frequency of c appeared in document d_j as follows (formula (2)):

$$freq_{cj} = \sum_{i=1}^{n} freq_{ij} \quad (2)$$

n represents the number of words belong to c in document d_j. So, the normalised frequency cf of c is (formula (3)):

$$cf = \frac{freq_{cj}}{\max(freq_{cj})} \quad (3)$$

Where the maximum of frequency is calculated over all concepts which are appeared in document d_j. Let N is the total number of documents and n_i is the number of documents which the concept c appears in. The *idf* for c is given following (formula (4)):

$$idf = \log(\frac{N}{n_i}) \quad (4)$$

So, we may calculate the *w(c),* the weight of c, using *cf*idf*, and it is given by formula (5):

$$w(c) = cf * idf \quad (5)$$

Combining the similarity function of concepts and weights of concepts, we have a function between to calculate the relatedness between documents and queries. Let *R (d, q)* denote the relatedness function for document d and query q, it is given by formula (6).

$$R(d,q) = \sum_{i=1}^{l} \max\{sim(c_i, c_j) \cdot w(c_j) \mid j = 1,2,...k\} \quad (6)$$

Here $c_i \in Q_c$, $c_j \in S_c$; $w(c_j)$ represents the weight of concept c_j belong to S_c. $l = |Q_c|$ and $k = |S_c|$. According to the relatedness function, the most related documents will be returned to the peer issuing the query.

6. Simulation experiment

In this section, we evaluate the keyword search mechanism proposed in this paper by simulation experiment. In order to analyze the retrieval performance, we ran a web crawler that visited the web pages on the Yahoo news and download the text and HTML files recursively. Our crawler downloaded about 10,935 HTML pages, covering 4 topics: Business, Sports, Science, and Technology. We develop a HTML parser using Java to clean HTML tags and extract plain text. We also develop a text parser using Java to eliminate the stop words and replace words by stems, adopting the algorithms introduced in [19]. We use Jena[20] to access the RDF/OWL representation of WordNet. *Jena* is a Java implementation for basic RDF handling. It aims at standard compliance and a friendly access from Java.

We generated concept set from WordNet for each text downloaded, and created index entries with all concepts of the set for this text. We use the synset URIs of WordNet RDF/OWL representation to denote concepts. We write index entries of texts to inverted index files based on concepts (concept index file). Each line in the concept index files represents an index

entry, and contains hash of an index concept, and a pointer list of documents which this concept appeared in. Each document pointer points the concept set of document and document metadata. Each line in the index files is showed as follows:

$< hash(\ synset\ URI\ ),\ \{doc_1\ pointer, doc_2\ pointer, \ldots, doc_n\ pointer\}>$

In order to compare concept index with existing keyword index scheme, we also construct inverted index files based on keywords (keyword index file) for these same texts. What contained of each line in the keyword index files are hash of a keyword and a pointer list of documents that contain this keyword. Each document pointer points document metadata. For similarity function *sim*, we set $\alpha = 0.2$ and $\beta = 0.6$ as used in [17].

We simulated search among peers of structure P2P network by search among the index files. So, we analyzed overhead and retrieval performance for a query by doing a search with keyword match in keyword index files and with concept match in concept index files respectively. Query overhead is the number of bytes transmitted when a user issues a query. The overhead to send the intermediate result list in the system from one peer to another is the main part of query overhead. Figure 3 gives the mean KB transmitted when a user issued a query using keyword match and concept match respectively. Figure 3 shows the query overhead of concept match is much lower than that of keyword match.

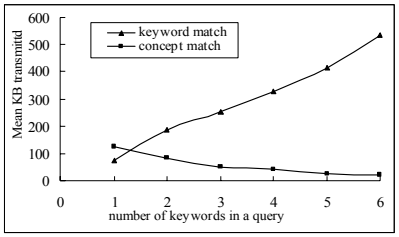

Figure 3. Query overhead

The query performance is evaluated using the standard information retrieval measures, precision and recall. Figure 4 shows the change of precision and recall of query using keyword match and concept match respectively. As shown in Figure 4, compared with commonly keyword match, concept match can significantly improve the precision and recall of query.

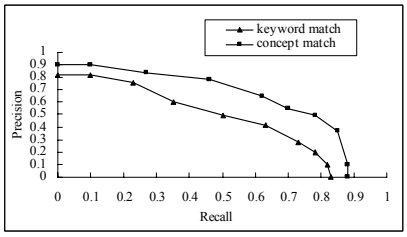

Figure 4. Precision-recall graph for query

7. Conclusions

In this paper, we present a scalable semantic search method over structure P2P network. We combine ontology with word sense disambiguation technology to determine the correct meaning of word. We construct a hybrid concept index distributing among peers of P2P network. The hybrid concept index avoids inverted lists join operation among peers. Also it transforms the keyword search to the concept matching of document and query. Simulation experiment shows that comparing to the keyword index, the retrieval performance of search with hybrid concept index is improved greatly, and generating lower traffic of network.

8. Acknowledgements

This paper is sponsored by NSF of China (No.90612001), Science and Technology Development Plan of Tianjin , (No. 043800311, 043185111-14) and Nankai University Innovation Fund and ISC.

9. References

[1] I.Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In Proc. of the ACM SIGCOMM ’01, 2001.

[2] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In ACM SIGCOMM, 2001.

[3] The Napster Homepage.: http://www.napster.com

[4] The Gnutella Homepage.: http://gnutella.wego.com

[5] Reynolds, P., Vahdat, A. Efficient peer-to-peer keyword searching. Technical Report 2002, Duke University, CS Department, Feb. 2002.

[6] Omprakash, D. Gnawali. A Keyword-set Search System for Peer-to-Peer Networks. MIT's thesis Lib, 2002.

[7] Jinyang Li, Boon Thau Loo, Joseph M.Hellerstein, M.Frans Kaashoek, David R.Karger, and Robert Morris. On the Feasibility of Peer-to-Peer Web Indexing and Search. In IPTPS, 2003.

[8] Miller,G. A. et al. Introduction to WordNet: An On-line Lexical Database, in the attached specification of WordNet 1.6,1993.

[9] Toan Luu, Fabius Klemm, Ivana Podnar, Martin Rajman, Karl Aberer. ALVIS Peers: A Scalable Full text Peer-to-Peer Retrieval Engine. In P2PIR'06, Arlington, Virginia, USA, 2006.

[10] Shuming Shi, Guangwen Yang, Dingxing Wang, JinYu, Shaogang Qu, and Ming Chen. Making Peer-to-Peer Keyword Searching Feasible Using Multi-Level Partitioning. In IPTPS, 2004.

[11] Nicholas J. A. Harvey, Michael B. Jones, Stefan Saroiu, Marvin Theimer and Alec Wolman. SkipNet: A Scalable Overlay Network with Practical Locality Properties. USITS'03, 2003.

[12] C. Tang and S. Dwarkadas. Hybrid global-local indexing for efficient peer-to-peer information retrieval. In NSDI, 2004.

[13]C.Tang, Z. Xu and M. Mahalingam. PeerSearch:Efficient Information Retrieval in Peer-to-Peer Networks[J].HPL - 2002- 198.2002.

[14] Gruber T R. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 1993. 5, 199～220.

[15] George A. Miller, Richard Beckwith, Christiane Felbaum, Derek Gross, and Katherine Miller, "Introduction to WordNet: An On-line Lexical Database", International Journal of Lexicography, Vol. 3, No. 4, 1990, 235 – 244.

[16]http://www.w3.org/TR/2006/WD-wordnet-rdf-20060619/.

[17] Yuhua Li, Bandar ZA, McLean D. An approach for measuring semantic similarity between words using multiple information sources. IEEE Trans. on Knowledge and Data Engineering, 2003,15(4): 871-882.

[18] M. Berry, Z. Drmac, and E. Jessup. Matrices, Vector Spaces, and Information Retrieval. SIAM Review, 41(2): 335–362, 1999.

[19] W.B. Frankes and R. Baeza-Yates. Information Retrieval: Data Structure and Algorithm. Prentice Hall, Englewood Cliffs, NJ, USA, 1992

[20] Brian McBride. Jena: Implementing the rdf model and syntax specification. 2001.