Adaptive Semantic Indexing of Documents for Locating Relevant Information in P2P Networks

Anupriya Elumalai[1] and Sriman Narayana[2]
[1]Department of Information Technology, Ibri College of Technology, Oman
[2]School of Computing Science and Engineering, VIT University, India

Abstract: *Locating relevant information in Peer-to-Peer (P2P) system is a challenging problem. Conventional approaches use flooding to locate the content. It is no longer applicable due to massive information available upfront in the P2P systems. Sometime, it may not be even possible to return small percent of relevant content for a search if it is an unpopular content. In this paper, we present adaptive semantic P2P content indexed system. Content indices are generated using topical semantics of documents derived using Wordnet ontology. Similarities between document hierarchies are computed using information theoretic approach. It enables locating and retrieval of contents with minimum document movement, search space and nodes to be searched. Results illustrate that our work can achieve results better than Content Addressable Network (CAN) semantic P2P Information Retrieval (IR) system. Contrary to CAN semantic P2P IR system, we have used content aware and node aware bootstrapping instead of random bootstrapping of search process.*

Keywords: *IR, semantic indexing, P2P systems, chord, concept clustering, lexical ontology, wordnet, semantic overlay network*

Received October 14, 2013; accepted July 24, 2014, published online January 14, 2015

1. Introduction

The Peer-to-Peer (P2P) technology which is the heart of cloud computing is gaining attention due to scalability, low cost, fault tolerance and self organizing nature increasing the possibility of effective P2P Information Retrieval (IR) systems [1, 2]. The regular storing, indexing and searching algorithms used on fixed infrastructures are not directly applicable to P2P systems due to issues like: The dynamic nature of configuration, missing global view and random distribution of documents without considering semantics [1]. Locating relevant content would be highly challenging in such an environment. Sometimes it may not be even possible to retrieve small percentage of relevant documents if it is not popular information.

The primary challenge in P2P IR system is to locate relevant document in accordance with query and retrieve those documents efficiently. Traditional search techniques in P2P systems use keyword based techniques and flooding, to locate the content in the system which is not good enough either to rank the relevance of documents or to locate the content within minimal search space. This led to semantic approach and also increased the significance of structured P2P systems [3, 4]. We propose a fairly novel adaptive semantic P2P content indexed system. The primary objective of our proposed system is to reduce the dimensionality of documents which would represent its semantics more precisely. The documents can then be organized and indexed in order to locate and retrieve documents more efficiently. Moreover, the inherent dynamic nature of P2P system requires more adaptable semantic indexing to address the search process irrespective of nodes joining and departing from the network.

The challenges to be addressed in order to realize the objectives of our proposed system are:

1. Reduce high dimensional document to low dimensional semantic space.
2. Cluster the documents according to semantics and organize them in semantic overlay network.
3. Index documents according to semantics therefore, they become cluster aware and node aware.

In order to reduce the dimensionality of documents, the documents are mapped to hierarchical structures in this paper using noun-to-noun (IS-A) relation. To obtain semantic hierarchy, we use Wordnet. Wordnet generates synsets. The notion behind is: synonyms are used to represent concepts. In this paper, we have used only hypernym and hyponym accounting to reason that they determine the semantic kernels of the document. This semantically disambiguates words within close proximity. Keywords are interpreted as synonyms that represent a specific concept [6]. The similarity between hierarchical structures can be obtained by comparing the semantics in common between two concepts in the hierarchies [12]. There are two approaches to calculate similarity between two concepts: Information theoretic based and conceptual distance based [4, 12]. The later requires more

structural information of taxonomy and may not be applicable to generic taxonomies. Moreover, it is originally not intended to compute similarity. Therefore, we adopted former method to compute similarity between concepts present in different documents. Two documents are similar if they share more common concepts. The more they share, the more they are similar. The concept hierarchies of two documents are accounted on similarity of information using modified Frequent Pattern (FP) growth algorithm. The documents repeating same pattern of concepts are grouped together based on proximity of documents. The patterns enable cluster awareness and the documents are semantically indexed across P2P overlay networks.

Conventionally the data items are distributed over P2P using Distributed Hash Table (DHT) such as in Content Addressable Network (CAN) and Chord [14, 16]. In this paper, we have used Chord to construct semantic overlay using cluster semantics in order to obtain abstraction of peers. This would identify similar clusters in nearby peers for efficient retrieval. We address the major challenges leveraging the inherent properties of clusters and semantic space. Exploiting the lower dimensional space, we have constructed semantic indices which enable cluster as well as node aware bootstrapping. Our cluster directed search reduces the search space substantially. We have developed P2P IR system prototype which represents documents as conceptual hierarchies. The documents are clustered in accordance with semantics and mapped on to nodes in Chord virtual ring.

The rest of this paper is organized as follows: Section 2 discusses the details of other's work related to our work. Section 3 provides overview of our system. Section 4 explains the semantic clustering and cluster key extraction. Section 5 provides information on P2P semantic indexing and adaptive indexing. Section 6 discusses on results and section 7 concludes the paper.

2. Related Work

P2P systems like napster use centralized indexing system. Such centralized systems suffer from one point failure and high load on index server leading to degradation in performance. Gnutella like systems use flooding which consumes large amount of network bandwidth. In order to probe minimum number of nodes, some kind of heuristics are to be employed to direct search query to fraction of nodes in the network. There are four kinds of such approaches: Random walk, use summary information, organize similar contents on neighboring nodes and organize groups to share common interest information.

Papers like Liv *et al*. [13, 17] discuss random walk and replication strategies. It has been inferred that random walk has proven to be more efficient than blind flooding. Also, they have found replication of indices on other nodes increases efficiency. Cuenca-Acuna *et al.* [4, 15] use bloom filters to get content summary of neighbors. In such case, the query is directed only to nodes with high probability of holding relevant contents. Schwartz [16] study organizing nodes into groups according to their content and association between nodes.

Papapetrou *et al.* [14] map each term into an ID and use term ID as key to store in DHT. It uses inverted list of terms to search documents with high matching query terms. The author has found that the algorithm has exponential growth proportional to input size. Hammouda and Kamel [7, 8, 9] explain collaborative clustering approach in which the summary of clusters are exchanged among the distributed nodes and query is directed to nodes with relevant documents or enrich the peer nodes with missing documents. Yan and Han [20] uses extended graph based substructures to reduce dimensionality. Except for the paper Cuenca-Acuna *et al*. [4], all other papers use plain keywords or keyphrases to search which may not be useful with huge information and large number of P2P nodes. The rationale behind our work is not to locate documents in match with query terms but to locate relevant documents with semantics and to direct search query only to candidate peers holding relevant content. This approach reduces search space substantially. Therefore, our work differs from others work by semantic indexing of content using frequently generated concept pattern and content directed bootstrapping of search process.

3. Overview of the Proposed System

In P2P systems, every node implies autonomy. This autonomy of peer nodes imposes difficulty in applying distributed clustering in P2P environment. In addition to that, peer nodes may join or leave the network. Consequently, the document clusters distributed in existing semantic overlay system should reflect the current setting without initiating clustering process from scratch. Thus, clustering should be carried out on local nodes and the semantic indices of clusters need to be distributed (indices should be shipped) in semantic overlay network which in turn would reduce communication overhead. Also, distributing similar clusters to neighboring peer nodes must be flexible and incremental. Traditional centralized clustering algorithms are based on global knowledge of data or schema. This meta information is further used for clustering the documents. However, the dynamic nature and missing domain knowledge imposes the difficulty of using predefined clusters and demands adaptive distributed mechanism to function with partial local knowledge obtained through coordination among peer nodes. Moreover, the assignment of clusters to fixed number of peer nodes may also require load balancing.

Our proposed system can be visualized as distributed catalog system based on structured P2P

system and it uses Chord for semantic overlay network. In this system, documents are clustered locally using topical semantics. FPs of concepts that represent the cluster semantics are generated. For each cluster, list of FPs form the cluster keys. i.e., semantic vector of that cluster. Similarly, there can be many clusters and their corresponding cluster keys. For each cluster c_i such that $c_i \epsilon C$, where $0< i< n$ where n is the number of clusters of node P_i. Cluster keys are presumed as data keys with their semantics. Hash value of cluster keys say for node P_i is calculated and the corresponding index is placed in the virtual node N_i. Clusters holding similar topics are mapped on to virtual chord ring such that the similar clusters (clusters with same semantics) are placed in neighboring peer nodes in the Chord ring. Each peer publishes its indices to the neighboring peers in the virtual ring. When a query is initiated at peer P_1, then the P_1 consults its finger table to find its neighboring peers containing clusters with similar semantics. The query vector Q is projected on attribute A: $A[Q]$. A semi join is performed at the neighboring peers say semantic vectors of P. $A[Q_m] \ltimes S_k[P_i]$ (say P_3, P_5, P_6 in Figure 1). The result from the neighboring peers form the candidate peers to be searched. The peer nodes joining and departing from network requires adaptable indexing for seamless functioning of P2P IR system. Therefore, we have extended cooperative mirroring for adaptive indexing. The peer n that wants to leave must handover its cluster indices to its successor in Chord address $(n + 2^{i+1}) \bmod 2^m$ and then leave the network. The cluster indices are replicated on several peers. Chord implementation exhibits robustness even in case of bad failures. At the worst cases, the successor peers take up the responsibility of maintaining cluster keys (index). We adopt cooperative mirroring scheme to adapt to changes in the network setting especially in terms of neighbors and cluster keys. Clustering is discussed in section 4 and P2P semantic indexing is discussed in detail in section 5.

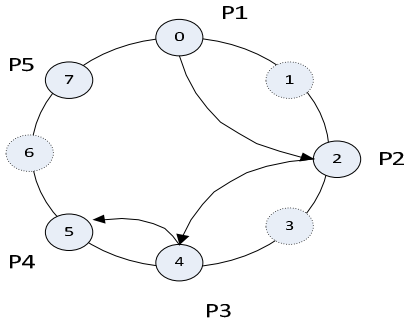

Figure 1. mapping of logical peers to physical peers.

4. Clustering

In P2P systems, clustering can be carried out in two ways [9, 10, 11]: Data items or indices of similar data items) can be clustered in such a way that similar data items (or indices) are placed in nearby peer nodes and peer nodes can be clustered so that the distance between the peer nodes are minimum; Data items (or indices) are stored in the semantic overlay network using (key, value) pairs using hash function. In the (key, value) pair, key refers to the path and value refers to single or summary of values (depends on domain) [15, 16]. If the hashing is augmented with order preserving, then similar clusters are stored in neighboring peer nodes. The curse of dimensionality reflects insufficient physical peer nodes to be mapped on to high dimensional content space and imbalance in index distribution. Moreover, range of successor peer nodes need to be searched to locate required content. Clustering of peer nodes reveals more natural approach such that the peer nodes can have the autonomy of storage, but it requires additional inter cluster and intra cluster routing mechanism. In reality, no single solution is found to be good. In this paper, we have adopted hybrid clustering approach, in which first documents are clustered according to semantics. Then, these cluster indices are distributed on peer nodes. These peer nodes are then mapped on to virtual ring. On need, semijoin is initiated at the peer nodes to retrieve candidate peers to be searched in order to locate the relevant content efficiently. This approach reduces the search space first at semantic level and then at the peer nodes level. Also, it eliminates the search in range of peer nodes on virtual ring preserving local storage autonomy, heterogeneity and topology.

4.1. Document Clustering and Cluster Key Extraction

The relationship between concepts of a document can be represented as hierarchical structure with directional edges eliminating loops [1, 16]. This can be visualized as Tree (parent to child) data structure, in which each node m_i in the tree represents a concept c_i and each edge e_{ij} represents relation between the nodes m_i and m_j. In reality, the hierarchical structures are huge and wide, but still each hierarchy includes distinct concepts and domain. The hierarchies are not exclusive, as one concept may be interlinked with another. The similarity between two concepts is measured with the information content value of the concepts that subsume them. The more they share, the more they are similar. Semantically, similarity between two documents can be formally defined as:

$$Similarity\,(c_1, c_2) == \begin{cases} max_{c \in Sup(c_1, c_2)} IC\,(c), c_1 \neq c_2 \\ 1, c_1 = c_2 \end{cases} \quad (1)$$

The clustering process is shown in Figure 2. It describes clustering and cluster key extraction. The notion behind cluster key extraction is to build better cluster representatives which aptly represent the cluster documents. We have adopted modified FP growth algorithm discussed in [7, 10, 14, 20] to mine frequent sub graphs which can represent the cluster documents semantically. The detailed description of information content based similarity computation and similarity histogram clustering process are discussed in our previous papers [2, 3]. In brief, it is discussed here to

get the overview of clustering. A Document Graph (DG) is generated for every document in the cluster using WordNet ontology [19]. Subsequently a Master Document Graph (MDG) is constructed with document graphs of cluster documents. From MDG, frequent sub graphs are mined using modified FP growth algorithm [20, 21]. It is a sense based approach to generate cluster representatives as it constructs super-ordinates or IS-A relationship to arrive at abstraction of a concept or group of words. Conditional pattern base are constructed followed by frequent repeating patterns. MDG will contain cumulative frequency corresponding to nodes.

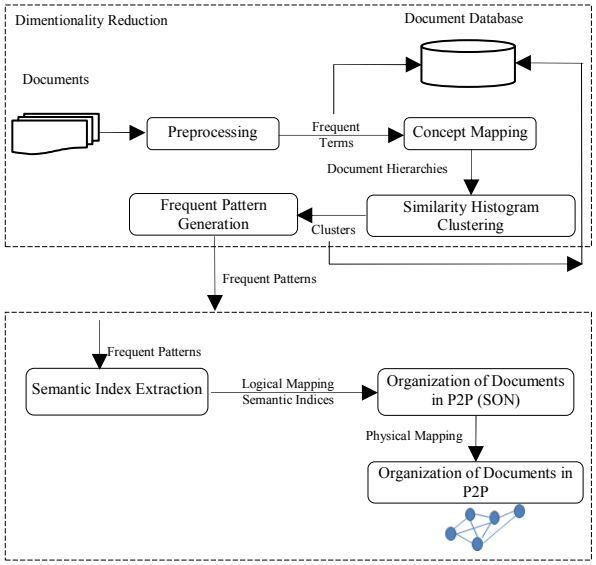

Figure 2. Adaptive semantic content indexing system.

A concept which occurs frequently in the cluster documents provides enough information to represent a cluster. The frequent subgraphs are semantic patterns of a cluster [17]. They are indexed as cluster indices. Each cluster represents one dimension (with dominating concept). Therefore, a node can have n clusters of d dimensions. In each dimension, there can be *m* documents. Say if the repeating pattern is {*cyclone*, *windstorm*, *weather_ condition*}, {*depression*, *cyclone*} in a cluster, then the topical semantics of the cluster it represents is {*cyclone*, *storm and weather_condition}*. This constitutes cluster key (or summaries of keys pertaining to a domain) and it may contain *m* documents in this dimension. A node *P* can contain *d* dimensions and semantic vectors corresponding to each dimension.

5. P2P Semantic Indexing

Query can be initiated from any peer node in the network. The documents in response to query need to be located at many peer nodes. Traditionally centralized or distributed indices (otherwise called as catalogs) are used to locate data in distributed systems. The indices are referred and the queries (or sub queries) are directed to peer nodes containing relevant data or documents. The documents from different peer nodes are then compiled as result and it is provided to the peer node which initiated the query. Constructing and maintaining these indices in P2P setting is highly challenging task. It requires indices to be updated frequently based on node joining or departing from network. Moreover, the numbers of peer nodes are steadily increasing which again requires indices to be highly scalable.

In general, three types of indexing approaches are followed in P2P environment: Centralized index, distributed index and no index. In centralized index, the indices (or catalog) are maintained completely on a single peer node. This limits the number of consulting requests leading to bottleneck at the central node or one point failure. Furthermore, maintaining multiple copies of centralized index resolves bottleneck and single point failure to some extend but fails to address frequent or large number of updates. In case of distributed index, the index is distributed fully over all nodes or only on super peers. Therefore, distributed indexing strongly depends on the overlay network topology: Structured or unstructured. In unstructured, the index is routed to nearby by peers and it ripples down the network. In reality obtaining complete information on peers is not possible. Boundaries or horizons are defined and index is routed to reachable peers. In case of structured P2P system, the information is assigned to peer based on hash value. In case of no index, the relevant content is located by flooding the request to neighboring nodes in the overlay network. It incurs large communication overhead.

Since, we have used Chord based overlay network, it would be more natural to adopt fully distributed index which is based on topical semantics. Clusters summaries are used to map documents on a virtual dimensional space with distributed hashing in overlay network. Each cluster key is added to the system using hash values. When a query is initiated, the semantics of the query is extracted to match with cluster keys found locally. Then, using the local catalog information, the candidate peers holding relevant clusters (relevant content) are retrieved. All retrieved peers of candidate peers are intersected to form final set peers to receive the query. The nodes or peers referred here are logical nodes. These logical nodes in the overlay network are further mapped on to physical nodes. Figure 1 illustrates an example of query routing in Chord extension. Solid line circles represent logical nodes in the Chord overlay network whereas the dotted line circle represents unassigned logical positions {1, 3, 6}. The labels next to circles are physical nodes{P_1, P_2, P_3, P_4, P_5}. The example is discussed with five logical nodes {0, 2, 4, 5, 7} organized in Chord ring. The Table 1 exhibits the index information that corresponds to each peer. P_1 knows that the successor of identifier 1 is peer 2, identifier of 2 is $P_2$from its finger table which are the identifiers and their successors. Algorithm 1 describes how P2P index is generated for cluster keys and how they are assigned to logical nodes. Algorithm

2 discusses how the query initiated at an arbitrary peer constructs candidate peers and contacts them to get relevant documents in accordance with query. For understanding the physical nodes are used in algorithms which will mapped on to logical nodes while processing.

Table 1. Semantic indices in chord and its extension.

Peer	Finger Table	Cluster Keys	Extended Cluster Key Summaries
P_1	1:2, 2:2, 3:4	C1	C1/d1{weather}, C1/d2{weather}
P_2	3:4, 4:4	C2	C2/d3{earthquake}
P_3	2:3, 2:4, 4:4	C3, C4	C3/d4 {news/election} C4/d5 {news/election}
P_4	5:5, 6:7	C5, C6	C5/d6 {cyclone} C6/d7{depression}
P_5	7:7	C7	C7/d8{entertainment/show}

Algorithm1: *P2P index generation and assignment*

Input: ip, port, cluster key
Output: identifier, value key

Method:
For each cluster key c_i^k *of node* P_j *where* $1<= j<=N$
//m: Number of identifier bits, c-cluster, i^{th} *key of cluster k is* C_j^k.
//N: Number of physical nodes, lgn_b *is the logical node*
//where $0 \le b \le (2^m-1)$
$id(P_j) = hash(ip(P_j), port(P_j))$
$id(c_i^k) = hash(c_i^k)$
Assign c_i^k *to* $lgn_{id(Pj)}$ $id(c_i^k)$
End For

Algorithm 2: *Locating relevant content and retrieval*

Input: query q_i, *initiating node* P_j
Output: candidate peers and results

Method:
For each query q_i *initiated by arbitrary peer* P_j
Compute query vector $q_i[\ ]$ *and process* $q_i[\ ]$ *in* P_j
Index $gP_j[qi] = q_i[\ ]$ *join* $d_k[P_j]$ *with max similarity*
Collect neighbors N [Pj] from finger table
//locate neighbors with similar content
// initially $C\{\} = \phi$
For each neighbor N_i *of* P_j
Ship q_i *index* $gP_j[q_i]$ *to neighbor* N_i *of* P_j *and* $C\{\}$
For each dimension d_k *in* N_i
Perform $q_i[\ ] \bowtie$ *vector* $d_k[N_i]$ *//semijoin*
Compute similarity of $q_i[\ ]$ *with* $d_k[\ ]$ *of* N_i
End for
If similarity found
Return reduct $q_i[\ ]$ *semijoin vector* d_k *with max similarity*
Return N_i
//indexes of neighbors N_i *of* P_j *with similar content*
Return $gPj[q_i] = gP_j[q_i] \cup gN_i[q_i]$
//candidate peer list
Return $C\{peer\text{-}lst\} = C\{peer\text{-}lst\} \cup N_i$
End if
End for
// visit only candidate peers to retrieve results
For each candidate peer N_i *with index* $gP_j[qi]$
Visit and perform $[q_i] \bowtie d_k[N_i]$
Return result to P_j *for* q_i
End for
End for (*query* q_i)

5.1. Adaptive Index Mechanism

The dynamic nature of P2P system requires update of index (adaptive index) to locate contents in other nodes P_{m-1} when a node P_m, departs from overlay network. When a node joins the network, clusters of local documents are constructed. The cluster key summaries are added to it using hash value generated by Chord. Thus, the new node will contain local clusters along with the other cluster summaries of the semantic overlay network. The finger table of peers are updated accordingly and propagated. When a node departs from the network, the clusters of the node and their semantics cannot be accessed. Before leaving the overlay network, the information containing references to other peers in the network are to be updated in the successor of the departing node. The successor node updates its own finger table and communicates to neighbors. Whenever there is query, the finger table is consulted to find any updates, if any node found to be departed or new node found to be joined, on the fly the peer nodes having similar clusters are found and updated.

For R keys and N nodes, each peer will hold B number of keys where $B=R/N$. By property of Chord, the keys will be evenly distributed with B number keys on nodes. Also, the keys are handed over to the successor i.e., $O(R/N)$ keys will be handed over. This causes latency in updating all peers in the overlay network. The latency in update would be high with large number of peers. Therefore, we have incorporated extension of cooperative mirroring scheme with Timer and Indices database shown in Figure 3. For every time unit t, the peer nodes in the overlay update indices to its neighbors. Since only finger tables are updated only in neighboring peers, the overhead caused will not affect the communication. The delay caused in propagating to all nodes in network can be reduced to minimum.

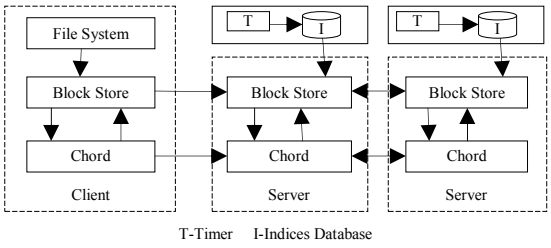

Figure 3. Cooperative mirroring extended to update indices for adaptiveness.

6. Results and Discussion

We built software prototype using Microsoft.Net with service pack 3 to implement DHT based P2P overlay network which is well accepted simulation pack to implement DHT based P2P overlay network. We have used the same to build clustering application on it. The system was developed on Pentium 2 GHZ computers with 2 GB RAM on windows OS in Visual Studio .Net IDE with C# language. The Text Retrieval Conference: TREC 7 and 8 [18] document set were used for

validating our algorithm. It is one of popular IR text corpus available on net. It consists of 528,543 documents accounting to size of 5 GB approximately with 500 topics as queries.

6.1. Average Neighbors and Routing Hops

For a n bit identifier space, we can have 2^n virtual nodes on Chord ring. The dimension of Chord is determined based on semantic dimensions. Semantic dimensions are larger compared to the number of physical nodes. Let d be number of semantic dimensions. These d dimensions if need to be partitioned along n physical node space ($d>=log_2\ k$) then the number partitions should be even and it will contain $log_2\ k$ neighbors where $k= 2^m$.

Figure 4 shows average number of neighbors with respect to number of nodes. It shows that the average numbers of neighbors accessed are linearly proportional to increase in number of nodes. Figure 5 shows the number of nodes with respect to average number of routing hops. In Chord, the routing is in $O(log\ N)$ where N is the number of nodes. It is understood from the Figure 5 that, the growth of average number of hops is also linearly proportional to increase in number of nodes.

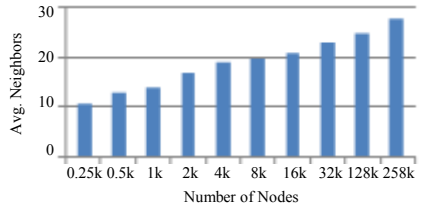

Figure 4. Average number of neighbors accessed by nodes.

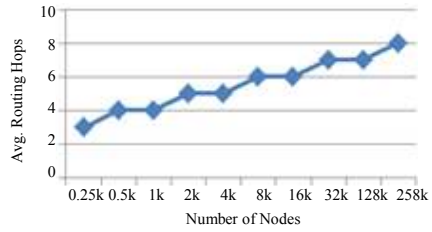

Figure 5. Average number of routing hops by nodes.

6.2. Even Distribution of Cluster Keys (Semantic Index)

In principle, if order preserving is hashing is used to allocate content, it may place similar contents in nearby nodes but the distribution of index may not be even. So, we have proposed cluster aware and node aware bootstrapping (basically content aware) in which first cluster keys are distributed evenly in virtual Chord ring. The TREC 7 and 8 corpus together are distributed evenly in x logical existing nodes out of 2^n nodes (maximum number of logical nodes). The cluster keys are distributed evenly in ring clockwise which is natural way of distribution over Chord but with semantics. The distributions of indices are compared with standard CAN semantic vector distribution. Table 2 displays distribution of indices in both systems. Table 2 values are derived as percentage of indices distributed over x% of peer nodes over total number of indices. In CAN semantic vector system the allocation of indices is random without including semantics for index generation i.e., like using document key. It is apparent that 5% of nodes carry 68% of indices displaying uneven distribution. In adaptive semantic P2P indexing system, the distribution of indices is balanced when compared to the base line CAN semantic vector system distribution.

Table 2. Distribution of indices over peers.

	CAN Semantic Vector System	Adaptive Semantic P2P Indexing System
% Nodes	% Indices	% Indices
5%	68%	16%
15%	71%	35%
25%	90%	42%
35%	96%	59%
45%	97%	66%
55%	97%	74%
65%	98%	80%
75%	99%	85%
85%	99%	89%

$$\%indices = \frac{Number\ of\ indices\ on\ x\ \%of\ peers}{Total\ number\ of\ Indices} \quad (2)$$

6.3. Number of Nodes Visited and Accuracy

The second objective of this paper is to reduce the number of nodes visited for a given query with better accuracy or same accuracy as results obtained earlier. We constructed a query set Q to evaluate number of nodes visited and number of documents returned. To evaluate our work, we gave same query set Q to both the systems. The results obtained are furnished in Table 3. The number of nodes visited indicates the resource consumption in proportion to visited nodes. 693 nodes are visited for approximately 2000 documents which ensure minimum nodes visited. The number of nodes visited in adaptive system is minimum compared to the CAN SV system. We refer to set of relevant documents as R for a query. We refer to set of returned documents using our work for the same query as S. Then, the accuracy A can be formally calculated as

Table 3. Result comparison.

CAN Semantic Vector System			Adaptive Semantic P2P Indexing System		
No. of Nodes Visited	No. of docs. returned	Accuracy in %	No. of Nodes Visited	No. of docs. Returned	Accuracy in %
102	17	78.2	88	13	84.1
108	29	81.1	90	22	86.0
116	46	83.6	94	41	87.4
141	91	83.9	136	83	88.2
199	185	83.3	189	164	88.6
231	260	86.3	216	246	89.1
343	501	87.4	311	480	93.0
481	1011	88.5	462	963	93.4
718	2210	86.9	693	1927	97.8

$$Accuracy\ A = \frac{|R \cap S|}{|R|} \times 100\% \quad (3)$$

Similarly, the number of documents retrieved is also more relevant in adaptive semantic P2P indexing

system when compared to CAN SV system. Results illustrates that adaptive semantic P2P indexing system performs better in terms of index distribution, number of nodes visited maintaining the accuracy of returned results.

7. Conclusions

The popularity of P2P technology has led to huge amount of information on these systems. Locating and retrieving relevant information effectively is a tedious task. It is also very difficult to find specific domain information for use. In this paper, we have presented adaptive semantic P2P content indexed system. Content indices are generated using topical semantics of documents derived using Wordnet ontology and information theoretic approach. Chord is used to build semantic overlay network. The semantic indices are cluster indices and they are distributed evenly among the peers. Furthermore, the peers can be grouped on the fly to retrieve documents. This approach of cluster aware and node aware bootstrapping enables to locate and retrieve relevant content with minimum semantic search space and minimum number of nodes. Results illustrates that adaptive semantic P2P indexing system performs better in terms of index distribution, number of nodes visited maintaining the accuracy of returned results compared to CAN based semantic IR system (in which the bootstrapping is at random point). Our contributions include:

a. Cluster Aware Bootstrapping and Node Aware: The search for content is not initiated at random point. Instead the search starts with the node where the relevant content is available based on semantic distribution of clusters.
b. Content Directed Search: The peer nodes containing relevant content are gathered first. (Information gathering). Then the search query is communicated only to those peers on the virtual ring. This reduces communication overhead and increases the response time.
c. Minimum Resource Consumption: The number of nodes visited indicates the resource consumption in the network.
d. Adaptive Indexing: The indices available are different peers are updated according to changing configuration of P2P overlay network incorporating extension of cooperative mirroring scheme.

Our paper addresses the challenges in locating relevant content in P2P systems. Our work can be applied in P2P environment and in semantic search engines.

References

[1] Al-Lahham Y. and Hassan M., "Scalable Self-Organizing Structured P2P Information Retrieval Model based on Equivalence Classes," *the International Arab Journal of Information Technology*, vol. 11, no. 1, pp. 78-86, 2014.

[2] Anupriya E. and Iyengar N., "Concept based Clustering of Documents with Missing Semantic Information," *in Proceedings of International Conference on Advanced Computing*, *Networking and Informatics*, Raipur, India, pp. 579-589, 2013.

[3] Anupriya E. and Iyengar N., "Peer-to-Peer Coordinated Virtual Clustering of Documents for Information Retrieval," *the International Journal of Information Processing and Management*, vol. 4, no. 6, pp. 86-98, 2013.

[4] Cuenca-Acuna F., Martin R., and Nguyen T., "PlanetP: Using Gossiping and Random Replication to Support Peer-to-Peer Content Search and Retrieval," *Technical Report*, Rutgers University, 2002.

[5] Eisenhardt M., Muller W., and Henrich A., "Classifying Documents by Distributed P2P Clustering," *in Proceedings of the 33rd Annual Meeting of the Society for Computer Science*, Frankfurt, Germany, pp. 286-291, 2003.

[6] Gale W., Church K., and Yarowsky D., "A Method for Disambiguating Word Senses in a Large Corpus," *Computers and the Humanities*, vol. 26, no. 5, pp. 415-439, 1992.

[7] Hammouda K. and Kamel M., "Distributed Collaborative Web Document Clustering using Cluster Keyphrase Summaries," *the Information Fusion Journal*, vol. 9, no. 4, pp. 465-480, 2008.

[8] Hammouda K. and Kamel M., "Hierarchically Distributed Peer-to-Peer Document Clustering and Cluster Summarization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 5, pp. 681-698, 2009.

[9] Hammouda K. and Kamel M., "Phrase-based Document Similarity based on an Index Graph Model," *in Proceedings of International Conference on Data Mining*, Maebashi, Japan, pp. 203-210, 2002.

[10] Han J., Pei J., Yin Y., and Mao R., "Mining Frequent Patterns without Candidate Generation: A Frequent Pattern Tree Approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53-87, 2004.

[11] Hassan M. and Abdullah A., "A New Grid Resource Discovery Framework," *the International Arab Journal of Information Technology*, vol. 8, no. 1, pp. 99-107, 2011.

[12] Lin D., "An Information-Theoretic Definition of Similarity," *in Proceedings of the 15th International Conference on Machine Learning*, Wisconsin, USA, pp. 296-304, 1998.

[13] Liv Q., Cao P., Cohen E., Li K., and Shenker S., "Search and Replication in Unstructured Peer-to-Peer Networks," *in Proceedings of the 16th International Conference on Supercomputing*, New York, USA, pp. 84-98, 2002.

[14] Papapetrou O., Siberski W., and Nejdl W., "PCIR: Combining DHTs and Peer Clusters for Efficient Full Text P2P Indexing," *Computer Networks*, vol. 54, no. 12, pp. 2019-2040, 2010.
[15] Rhea S. and Kubiatowicz J., "Probabilistic Location and Routing," *in Proceedings of 21st Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, USA, pp. 1248-1257, 2002.
[16] Schewartz M., "A Scalabale, Non Hierarchical Resource Discovery Mechanism based on Probabilistic Protocols," *Technical Report*, University of Colorado, 1990.
[17] Tang C., Xu Z., and Mahalingam M., "PSearch: Information Retrieval in Structured Overlays," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 89-94, 2003.
[18] Text Retrieval Conference (TREC)., available at: http://trec.nist.gov, last visited 2014.
[19] WordNet-Princeton University., available at: http://wordnet.princeton.edu/wordnet, last visited 2014.
[20] Yan X. and Han J., "GSpan: Graph-Based Substructure Pattern Mining," *in Proceedings of IEEE International Conference on Data Mining*, Maebashi, Japan, pp. 721-723, 2002.
[21] Zhong N., Li Y., and Wu S., "Effective Pattern Discovery for Text Mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 1, pp. 30-44, 2012.

Anupriya Elumalai received her Bcs of engineering from Faculty of Computer Science and Engineering, Madras University in 1997 and MS degree of technology in computer science and engineering from VIT University in 2004. Currently, she is working for Information Technology Department, Ibri College of Technology, Sultanate of Oman. She is pursuing her research in School of Computing Science and Engineering, VIT University, India. Her research interests include peer-to-peer data management, data mining, knowledge discovery from text data and information retrieval. She has 15 publications in Journals and Conferences to her credit. She is a member of ISTE, AIENG and IEEE.

Sriman Narayana recived his MS degree in applied mathematics, ME degree computer science and engineering PhD degree in applied mathematics. Currently, he is Director of Periyar EVR Central library and also Senior Professor at the School of Computing Science and Engineering at VIT University, India. His research interests include: Distributed computing, information security, electronic and mobile commerce applications, intelligent computing and fluid dynamics (porous media). He had 26 years of teaching and research experience with a credit of nearly 145 publications in reputed International Journals and Conferences. He has authored/co-authored several textbooks/learning materials for the student community. He chaired many International Conferences, delivered Key note/Invited/Guest/ Technical lectures, served as PC Member/Reviewer. He is Editor in Chief for International Journal of Software Engineering and Applications(IJSEA) of AIRCC, and Editorial Board member for International Journals like IJConvC (Inderscience -China), IJCA (USA) etc.