

# Emerging Semantic Communities in Peer Web Search

R. Akavipat, L.-S. Wu, F. Menczer  
{rakavipa, lewu, fil}@indiana.edu  
Department of Computer Science  
School of Informatics  
Indiana University  
Bloomington, IN 47408, USA

A.G. Maguitman  
agm@cs.uns.edu.ar  
Departamento de Ciencias  
e Ingeniería de la Computación  
Universidad Nacional del Sur  
8000 Bahía Blanca, Buenos Aires, Argentina

## ABSTRACT

Peer network systems are becoming an increasingly important development in Web search technology. Many studies show that peer search systems perform better when a query is sent to a group of peers semantically similar to the query. This suggests that semantic communities should form so that a query can quickly propagate to many appropriate peers. For the network to be functional, its dynamic communication topology must match the semantic clustering of peers. We introduce two criteria to evaluate a peer search network based on the concept of *semantic locality*: first, the “small-world” topology of the network; second, we use *topical semantic similarity* to monitor the quality of a peer’s neighbors over time by looking at whether a peer chooses semantically appropriate neighbors to route its queries. We present several simulation experiments conducted with different peer search algorithms on our peer Web search system, 6S. The results suggest that 6S, despite its use of an unstructured overlay network; can effectively foster the spontaneous formation of semantic communities through local peer interactions alone.

**Categories and Subject Descriptors:** H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; H.3.4 [Information Storage and Retrieval]: Systems and Software

**General Terms:** Performance, Algorithm, Experimentation.

**Keywords:** Peer search, semantic locality, small-world networks, topical semantic similarity, global coherence, coverage.

## 1. INTRODUCTION

Distributed systems have emerged as part of the solution to the scalability limitations of centralized search engines. Even “traditional” search engines employ distributed and parallel systems to handle the massive computational and storage requirements of indexing, retrieval and ranking. Furthermore, as search becomes more prevalent at the desktop level, it is easy to foresee a near future when, in addition to public Web servers, users will make portions of the files indexed in their computers available to others via the Internet. As a result, peer network architectures are receiving increasing attention in the context of Web search technology.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

P2PIR '06, November 5–11, 2006, Arlington, Virginia, USA.  
Copyright 2006 ACM 1-59593-433-2/06/0011 ...\$5.00.

There are many aspects of peer-to-peer application performance, such as the degree of fault tolerance, scalability, quality of service, and network traffic. In this paper, we propose a different factor that can also predict the performance of a peer search application, the existence of *semantic communities* in peer systems. To achieve good results for a text query, a peer search system must try to predict which peers are best suited for the query. The best candidates for the query are the peers whose content is semantically closest to the query. Several studies confirm this observation [8, 27, 31]. They show that when peers are grouped by their semantic similarity, the performance of the system increases.

Therefore, in this paper, we argue that the emergence of semantic communities is another important feature in peer search systems. In particular we focus on 6S, a peer search system under development by our group, and show that semantic communities can emerge even under the unstructured peer network model used by 6S. To evaluate the emergence of semantic communities, we propose measures based on two tests:

- **Small-world network topology.** Two network statistics, the *diameter* and the *clustering coefficient*, are used as indicators of the “small-world” topology of the underlying peer-to-peer network [33]. A small diameter corresponds to a small separation between peers, while a high clustering coefficient signals tight communities. Intuitively, a network satisfying these properties would allow peers to reach each other via short paths while maximizing the efficiency of communication within semantically clustered communities.
- **Topical Semantic Similarity.** We assume for evaluation purposes that queries and peers are associated with a subset of topics from the Open Directory Project<sup>1</sup> (ODP). A measure of semantic similarity between topics in the ODP [20] is used to evaluate query routing algorithms. The quality of a peer’s neighbors is monitored over time by looking at whether a peer chooses “semantically” appropriate neighbors to route its queries.

## 1.1 Background

A *peer-to-peer* (P2P) computer network relies on the computing power and bandwidth of the participants in the network rather than concentrating it in a relatively few servers. The most popular use of a P2P network is file sharing. Applications such as Gnutella, BitTorrent and KaZaa [3] allow peers to share content files without dedicated servers or large bandwidth to support the whole community. The P2P file sharing application provides an alternative for content distribution by trading the speed and reliability of ded-

<sup>1</sup><http://dmoz.org>

icated servers for ease of sharing, lower cost, fault tolerance, and lower bandwidth requirement.

In a similar way as P2P file sharing applications are used to facilitate content distribution, P2P applications can be developed to facilitate Web search. There is a wide variety of peer-based search applications. For example, a model proposed by the YouSearch project is based on maintaining a centralized search registry for query routing (such as Napster), while providing the peers with the capability to crawl and index local portions of the Web [5]. A completely decentralized approach is illustrated by the Gnutella model, in which queries are sent and forwarded blindly by each peer. Another system, NeuroGrid [11], employs a learning mechanism to adjust metadata describing the contents of nodes. A similar idea has been proposed to distribute and personalize Web search using a query-based model and collaborative filtering [23]. An intermediate approach between the flood network and the centralized registry is to store index lists in distributed, shared hash tables [28]. In pSearch [29] latent semantic analysis [9] is performed over such distributed hash tables to provide peers with keyword search capability. Another alternative are hybrid peer networks, where multiple special directory nodes (hubs) construct and use content models of neighboring nodes to determine how to route query messages through the network [17].

6S [35] uses the same idea of content based models of neighboring nodes, but without assuming the presence of special directory hubs. Each peer is both a (limited) directory hub and a content provider; it has its own topical crawler and local search engine. Queries are first matched against the local engine, and then routed to neighbor peers to obtain more results. Peers learn from their interactions how to route queries to semantically related nodes. While traditional search engines such as Google and Yahoo provide access to very large document collections, the 6S P2P Web search application provides a complementary way for users to actively and collaboratively share their own document collections. However, the 6S framework allows to naturally include traditional search engines as peers; if the system works, such peers would quickly emerge as reliable, trustworthy, and general authority nodes.

Adaptive query routing and the semantic locality of peers have also been explored in the file sharing domain [12, 30]. SON [8] employs an explicit grouping system for peers. Peers form an overlay network of semantic groups according to their document collections. A classification hierarchy is used as the basis for this overlay network. Each peer and query is classified into one or more leaf concepts in this hierarchy according to its content. Then peers with semantically similar contents (i.e. belonging to the same concept) will be grouped together. Peers can join more than one group. A query is sent to groups that have higher probability to answer it. Then the query is propagated only inside those groups. Despite the drawback of requiring extensive user intervention in query classification and determining the number of groups, the study shows that search results can be improved by such a semantic grouping of peers. Whether semantic communities are formed explicitly as in SON or implicitly as in 6S, a related problem is how to efficiently discover these peer communities [13].

## 2. SEMANTIC COMMUNITIES

In this section we propose two methods that can be applied in the evaluation of the semantic community aspects of a peer search system. The first method looks at the emergent topology of the P2P network. We propose that a good topology is one that favors efficiency by making it possible for a query to reach a target in few steps, without imposing a large bandwidth load on the system. The second method analyzes the emergent network from a semantic per-

spective. Ideally, peers will submit their queries to other peers that specialize on the topic of the query. We argue that a good network topology is one in which there is a high degree of semantic similarity between the topics of the peers that talk to each other, because this would allow queries to quickly propagate among relevant peers as soon as one of them is reached.

### 2.1 Small-world Network Topology

Search efficiency is a major issue in any Web application. The query delivery mechanism of a peer network will have a tremendous impact on its efficiency. On one hand we want queries to reach a good target in a small number of steps, while on the other hand we want a non-congested network [27, 1, 31, 7]. There is a trade-off between the short search paths of random networks and the prevention of congestion provided by local network structure.

In an adaptive collaborative peer network system, peers attempt to retrieve quality results by sending queries only to a few good neighbors. This interaction between peers determines the topology of the network. A good topology should allow for any two peers to reach each other via a short path while maximizing the efficiency of communication within clustered peer communities. As a consequence, analyzing the topology of the peer network is an implicit way to measure the efficiency of the peer-based search application.

This special peer gathering property of the collaborative peer network should lead to an emergent clustered topology. In such a topology neighbor communities will tend to form according to clusters of peers with shared interests and domains. It follows that the ideal topology for such a network would be a “small world” [33].

To determine if the peer network topology exhibits small-world properties, we have to measure the network’s *clustering coefficient* and *diameter*. The clustering coefficient for a node is the fraction of a node’s neighbors that are also neighbors of each other. This is computed in the directed graph based on each peer’s  $N_n$  neighbors, with a total of  $N_n(N_n - 1)$  possible directed links between neighbors. The overall cluster coefficient  $C$  is computed by averaging across all peer nodes. The diameter  $D$  is defined as the average shortest path length  $\ell$  across all pairs of nodes. Since the network is not always strongly connected, some pairs do not have a directed path ( $\ell = \infty$ ). To address this problem, we use the harmonic mean of shortest paths:  $D = N(N - 1) / \sum_{i,j} \ell_{ij}^{-1}$  where  $N$  is the number of nodes. The diameter  $D$  thus defined can be computed from all pairs of nodes irrespective of whether the network is connected.

If the diameter of a network remains similar to that of a random network, while the clustering coefficient increases, then we have an indication of the emergence of a small-world topology [33]. This result would point toward the existence of several clusters in the emergent network. We postulate that such a network provides a better environment in terms of search length and network congestion than either a random network (low  $D$  and low  $C$ ) or a regular network (high  $D$  and high  $C$ ). This test can be applied to study the evolution over time of a random network toward a small-world network topology, as well as to compare the topology of any two peer networks.

### 2.2 Topical Semantic Similarity

The test described above can help evaluate the performance of a peer search application in terms of efficiency. However, it does not provide insight into the way peers interact from a semantic perspective. Here, we propose to look at the topics associated with peers and queries with the purpose of evaluating how they affect the communication patterns in the network.

Adaptive peer-based Web search systems allow peers to form communities without centralized control. As a consequence, a peer

discovers new nodes through its current neighbors rather than by contacting some central hubs. Peers progressively learn and store knowledge about other peers with a view to their potential for answering prospective queries. This enables peers to learn the dynamic properties of the network, which include the network topology and peers’ knowledge.

A good learning algorithm would help peers predict which other peers have the knowledge required to respond to a query. Therefore, we hypothesize that in a well adapted network: (1) the topical semantic similarity between neighbors will increase as the network adapts; (2) the similarity between each query and the peers that the query reaches will also increase over time. Indeed, these two hypotheses point to the essential idea behind adaptation in a peer-based search system: a network topology with *semantic locality* should emerge as peers learn about each other.

To evaluate the effectiveness of the adaptive mechanism, measures of semantic similarity between peers and between queries and peers are needed. Semantic similarity can be used to describe the degree of relatedness between the meanings of topics, as perceived by users. Measures of semantic similarity based on taxonomies are well studied [25, 16]. Recently the tree-based information-theoretic similarity measure has been extended to general ontologies, where both hierarchical and non-hierarchical components are considered [21, 19]. This measure has been successfully applied to the ODP graph, a human edited directory of the Web that classifies millions of pages into a topical ontology. In particular, if peers and queries are associated with topics in the ODP graph, we can measure the semantic similarity between two peers or between a peer and a query by computing the semantic similarity between the corresponding topics. In § 4 we describe how peers and queries can be mapped into ODP topics; note that the use of a directory such as the ODP is only necessary for evaluation purposes and not require for the normal use of the peer search system.

### 3. 6SEARCH FRAMEWORK

A detailed description of 6S is out of the scope of this paper and can be found elsewhere [35]. Here we just sketch the 6S network protocol, its neighbor management mechanism and three different algorithms that 6S can use to route its queries.

#### 3.1 6S Protocol

The 6S peer network protocol acts as an application layer between the search engine and the network (TCP/IP) layer. The application also interfaces with the network using the HTTP protocol for crawling the Web. The 6S peer network layer provides the means to find results (hits) by querying the indexes built by peer search engines. When the user submits a query to the application, it can retrieve hits from its local index database and augment the results by searching the peer network for additional hits.

The 6S protocol has the following properties: (1) peers are independent; (2) a peer can enter and leave the network at any time; (3) a peer should not be overwhelmed by other peers; (4) a query should not be propagated indefinitely; (5) a peer may choose not to forward or respond to some queries; and (6) the architecture should make it difficult to create denial of service attacks.

#### 3.2 Neighbor Management

The 6S system is designed not to have peers aggressively flooding the network looking for other peers unless it is necessary to do so, such as when a peer enters the network for the first time or when no known peer is available. Normally a peer discovers new peers through its current neighbors.

The 6S protocol gives each peer a fixed number of slots for

neighbors,  $N_n$ , depending on their bandwidth and computational power to process neighbor data. A peer will search for new peers when its neighbor slots are not full or when it wants to find better neighbors than the currently known peers.

Many neighbor management algorithms in the P2P literature require peers to send update messages in order to maintain valid network information when peers leave the network. In contrast, a 6S peer does not need to send any message when it wants to leave the network because the query routing algorithms (described next) update information about neighbors based on queries and responses.

### 3.3 Adaptive Query Routing

To route queries appropriately, each peer will learn and store profiles of other peers. A neighbor profile is the information a particular peer maintains to describe its knowledge about what that neighbor stores in its search engine index. By adapting profile information, peers try to increase the probability of choosing the appropriate neighbors for their queries. Any type of peer learning algorithm can be plugged into 6S for adaptive query routing. We describe next three algorithms used in our simulations.

#### 3.3.1 Random-Known Algorithm

As a baseline we implemented a random query routing algorithm as used by Gnutella. This algorithm implements a trivial mechanism in which queries are routed to random neighbors. Peers learn about the existence of new neighbors through their known neighbors. Then a peer randomly chooses  $N_n$  among its known peers to send/forward queries. Although this algorithm does not use any neighbor selection scheme for query routing, peers are added to the list of known peers as they respond to queries. Thus a peer still learns and stores some minimal knowledge about the network.

#### 3.3.2 Greedy Learning Algorithm

The greedy learning algorithm is a simple adaptive algorithm to manage neighbor information and to use such information to dynamically select neighbors to query. Each peer maintains a peer profile matrix  $W$  in which rows correspond to terms and columns to peers. Thus, the value  $w_{i,p}$  in  $W$  is the importance of term  $i$  as a descriptor of the profile of peer  $p$  ( $p = 1, \dots, N_k$ ).

When peers get responses from neighbors (and neighbors’ neighbors), the responses are evaluated and used to update the profile of each known peer. The scores of hits received from each neighbor are compared with local hit scores. If a score of any neighbor hit is better than at least one of the top  $N_n$  local scores, the query keywords are added to the neighbor profile. The matrix  $W$  is updated to reflect the highest hit score returned by each such neighbor.

To select neighbors for routing a query  $Q$ , the similarity  $\sigma(p, Q)$  between each known peer  $p$  and the query is computed as follows:

$$\sigma(p, Q) = \sum_{i \in Q} w_{i,p}.$$

The top  $N_n$  ranked among known peers are selected as neighbors and sent the query.

#### 3.3.3 Reinforcement Learning Algorithm

Interactions with peers reveal information of varying reliability. For example, a direct response to a query is telling about a peer’s knowledge with respect to that query, but may also reveal (less reliable) information about the peer’s knowledge relative to other queries. We want to capture all available information in profiles, but must discriminate information on the basis of its reliability. This ability to reflect varying degrees of reliability is implemented in the reinforcement learning algorithm, where each peer maintains

two profile matrices,  $W^f$  and  $W^e$  for *focused* and *expanded* information, respectively. Each profile matrix has the same structure as the profile matrix used in the greedy learning algorithm.

**Focused profile:** weights  $w_{i,p}^f$  are initially updated based on  $p$ 's response to a neighbor profile request, and successively updated through query-response interaction—namely for terms  $i$  in queries submitted or forwarded to  $p$ . Based on the comparison of the incoming hits with its local hits for a query  $Q$ , a peer makes an assessment about  $p$ 's knowledge with respect to terms  $i \in Q$ .

**Expanded profile:** weights  $w_{j,p}^e$  are updated through query-response interaction analogously to the focused profile, but for terms  $j \notin Q$  that co-occur with terms  $i \in Q$  in a hit page  $d$  returned by  $p$ , such that  $j$  has a higher term frequency:  $TF(j, d) > \max_{i \in Q} TF(i, d)$ . If a certain set of documents is a good response for a certain query, then it may as well be a good response for queries that are well represented in the set. By this query expansion, we expect to speed up neighbor learning.

Upon arrival of a query response, a peer uses the following soft update rule to modify the weights of the query terms in the neighbor profile matrices:

$$w_{i,p}(t+1) = (1 - \gamma) \cdot w_{i,p}(t) + \gamma \cdot \left( \frac{S_p + 1}{S_l + 1} - 1 \right)$$

where  $t$  is a time step,  $S_p$  and  $S_l$  are the average scores of  $p$ 's hits and the local hits respectively in response to the query  $Q$ , and  $\gamma$  is a learning rate parameter ( $0 < \gamma < 1$ ). The terms  $i$  subject to this learning rule depend on  $Q$  and the profile matrix (focused or expanded) as described above.

To route a new query  $Q$ , known peers are ranked by similarity  $\sigma$  computed as follows:

$$\sigma(p, Q) = \sum_{i \in Q} \left[ \alpha \cdot w_{i,p}^f + (1 - \alpha) \cdot w_{i,p}^e \right]$$

where  $\alpha$  is a reliability parameter, typically  $0.5 < \alpha < 1$  to reflect higher confidence in focused profile weights as they come from direct responses to queries. The top  $N_n$  ranked known peers are selected to send/forward  $Q$ .

## 4. EXPERIMENT METHODOLOGY

We ran six computer simulations of 6S. Each simulation implemented one of the three routing algorithms and one of two scenarios for the assignment of local queries to peers (to be described in detail in § 4.3). In our simulations we modeled synthetic users and ran their queries over real indexes obtained from actual Web crawls.

### 4.1 Measuring Quality of Results

Small-world properties and topical semantic similarity are important aspects of a P2P search network, but do not reveal whether the search results obtained by a peer are relevant. Existing evaluation methodologies for Web search in general, and distributed search in particular face several technical difficulties. Well established IR approaches to evaluate the quality of results are based on precision, recall, and many related measures [4]. Precision is the fraction of retrieved documents which are relevant, while recall is the fraction of relevant documents which have been retrieved.

Computing these measures requires access to relevance assessments. Ideally, humans would provide assessments of document relevance to a given query. This is possible for a small corpus,

where relevance judgments can be based on an exhaustive examination of the document collection. However, for a large and dynamic corpus, such as the Web, this procedure becomes infeasible. An approach to overcome this difficulty is to use standard test collections, such as those provided by the Text Retrieval Conference (TREC) [14, 18]. These collection provide queries and relevant sets. However, for the present evaluation we need a sufficient number of queries so that the peers in the network have an opportunity to learn about each other. TREC collections provide at most a few hundred queries, while in the simulations described here we need several thousand queries from several hundred distinct topics.

Other techniques, such as the one proposed in [2], rely on users' assessments of term relevance to topics rather than document relevance to queries. Document-query relevance is automatically inferred based on the available term-topic relevance assessments. This approach facilitates result evaluations, but it still requires humans to provide topic assessments for a large amount of term-topic pairs. Many other techniques for coping with incomplete relevance information have been proposed and applied with varying success (e.g. [26, 6, 34]); measuring precision and recall for the Web domain remains a challenge.

In the face of this limitation, we turn to two novel criterion functions for evaluating retrieval performance: *global coherence* and *coverage* [15]. These two functions generalize the well known IR measures of precision and recall. However, in contrast to precision and recall, the measures of global coherence and coverage do not require that all relevant resources be precisely identified. Instead, these measures are applicable as long as an approximate description of the potentially relevant material is available.

Let us review the definitions of global coherence and coverage. Assume  $R = \{r_1, \dots, r_m\}$  is a set containing approximate descriptions of potentially relevant material, where each  $r_i$  is a collection of keywords. These can be extracted from any (sub)set of relevant pages. Let  $A = \{a_1, \dots, a_n\}$  be the set of retrieved resources, with  $a_i$  also represented as a collection of keywords. A measure of *similarity* between a retrieved resource  $a_i$  and a relevant resource  $r_j$  can be computed using the *Jaccard coefficient*:

$$\text{Similarity}(a_i, r_j) = \frac{|a_i \cap r_j|}{|a_i \cup r_j|}$$

Then, the *accuracy* of resource  $a_i$  in  $R$  is defined as follows:

$$\text{Accuracy}(a_i, R) = \max_{r_j \in R} \text{Similarity}(a_i, r_j).$$

The accuracy of a retrieved resource  $a_i$  provides an estimate of the precision with which the keywords in  $a_i$  replicate those of relevant resources. Once the accuracy of each retrieved result has been computed, it can be used to obtain a measure of global coherence as follows:

$$\Phi(A, R) = \frac{\sum_{a_i \in A} \text{Accuracy}(a_i, R)}{|A|}. \quad (1)$$

The global coherence function measures the degree to which a retrieval mechanism succeeded in keeping its focus within the theme defined by a set of relevant resources. This is similar to the IR notion of precision, except that it is based on a less restrictive notion of relevance: by using a measure of accuracy instead of considering exact matches it is possible to overcome the drawback of binary classification of relevancy.

A high global coherence value does not guarantee acceptable retrieval performance. For example, if the system retrieves only a single resource that is similar to some relevant resource, the global coherence value will be high. Because search mechanisms should also maximize the number of relevant resources retrieved, a coverage factor is introduced to favor those strategies that retrieve many

resources similar to a target set of relevant resources. A criterion function able to measure coverage is defined as a generalization of the standard IR notion of recall:

$$\begin{aligned}\Psi(A, R) &= \frac{\sum_{r_i \in R} \text{Accuracy}(r_i, A)}{|R|} \\ &= \frac{\sum_{r_i \in R} \max_{a_j \in A} \text{Similarity}(a_j, r_i)}{|R|}.\end{aligned}\quad (2)$$

A performance evaluation based on coverage and global coherence can partially overcome the incomplete relevance information problem if an approximate description of the potentially relevant material for a query is available. A simple way to construct a test set is by taking advantage of the information available in the form of URL descriptions in the ODP graph.

## 4.2 Query Generation

Each peer in our experiment was assigned 10 queries. Because our evaluation framework requires that queries be associated with topics, we implemented a procedure to automatically generate topical queries for each peer. A good query for a topic should be such that it produces quality results when submitted to a search engine. We use the measures of global coherence and coverage defined above to automatically assess the quality of a query. The following procedure was used to construct 10 high-quality queries for each topic  $t$ :

1. Consider the set  $U_t$  containing the URLs indexed in  $t$  or in any subtopic of  $t$ .
2. Use the ODP description of each URL  $u \in U_t$  to create candidate queries of length 2. Each query is constructed by using all pairs of consecutive words found in the descriptions, after stop word filtering.
3. Submit each candidate query  $q$  to Google and use the top 10 hits of each query to construct the answer set  $A_q$  for that query. Discard queries that receive less than 10 hits. The answer set  $A_q$  is a set of collections of terms, each collection created from the terms found in the snippets returned by Google as a response to query  $q$ .
4. Use the ODP description of each URL  $u \in U_t$  to construct the relevant set  $R_t$  for topic  $t$ . The relevant set  $R_t$  is a set of collections of terms, each collection created from the terms found in the URL descriptions.
5. Compute global coherence and coverage using Eqs. 1 and 2 for the answer set  $A_q$  with respect to the relevant set  $R_t$ .
6. Compute the score of each query as the harmonic mean of its global coherence and coverage measures:

$$F_1 = \frac{2 \cdot \Phi \cdot \Psi}{\Phi + \Psi}.$$

7. Rank all candidate queries by the score  $F_1$  and use the top 10 as final queries for topic  $t$ .

We contend that the queries constructed using the above procedure are good representatives of their corresponding topics.

## 4.3 Simulation Setting

The simulated peer-based search network contained 500 synthetic users, each associated with a unique topic. The topics were randomly selected from the third level of the ODP hierarchy and each topic contains at least ten URLs. Since each of the users had 10 local queries, we used a total of 5000 distinct queries. The peer network was initialized as a random *Erdos-Renyi* graph, i.e., each

peer was assigned 5 random neighbors drawn from a uniform distribution.

Each peer ran its topical crawler to collect documents and populate its index. For the topical crawler, we used a *best-N-first* search algorithm described in [22], which has been proven very effective against a number of crawling algorithms. The crawler was given a small set of topic keywords and a set of seed URLs to start from. The topic keywords were obtained from the two most specific labels used in ODP to identify that topic (e.g., “math” and “software”) were used as topic keywords for “Top/Science/Math/Software”). We used all the URLs indexed in the ODP topic as seed URLs. The topical crawler was run offline to harvest around 2,500 pages for each peer. The Nutch package<sup>2</sup> was then used to index these pages and build each peer’s search engine.

Two different scenarios were considered to run our simulations. In the first scenario, each peer was assigned 10 local queries from the same topic as itself. In the second scenario, each peer was assigned 10 local queries from randomly chosen topics different from its own. We refer to the two cases as *in-topic* and *off-topic* scenario, respectively.

Our simulation programs took a snapshot of the network at every time step. In a time step all the peers finish processing their buffered incoming messages and sending their outgoing ones. This may include generating local queries, forwarding other peers’ queries and responding to queries received from other peers. Each of our simulations was distributed over five 3.2 GHz hyper-threading P4 Linux machines, each running 100 peers. A complete simulation run took approximately 24 hours.

## 5. EMERGENT COMMUNITIES

As described in § 2, the emergence of semantic communities is determined from the small-world topology coupled with the topical semantic locality among peers within communities. We now discuss our findings relative to each of these criteria.

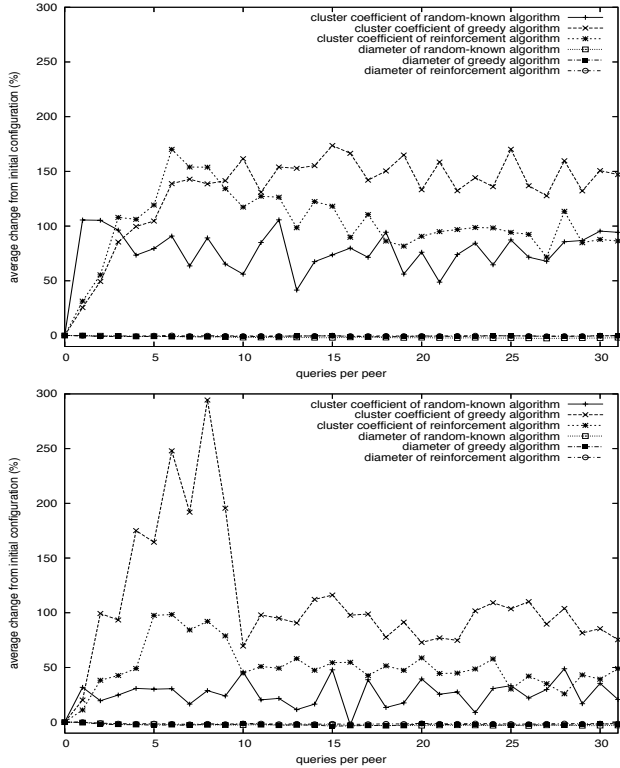
### 5.1 Small-world topology

As discussed in § 2.1, the topology of the peer network is a crucial factor determining the efficiency of a peer-based search system. It is expected that over time a good algorithm for query routing will transform the topology toward a small-world network. To verify this transformation, we monitored the changes in the clustering coefficient and diameter of our simulated networks. Figure 1 shows plots for these two quantities as a function of the number of queries sent by each peer for the three learning algorithms discussed in § 3.3.

For all cases, the initial network has a low clustering coefficient and a small diameter as expected in a random graph. As the network evolves the clustering coefficient increases very rapidly and significantly while the diameter remains almost unchanged. This points to the emergence of a small-world network topology, providing an efficient environment for P2P communication.

In both plots we observe that the random-known algorithm is the least effective in forming peer communities. The clustering coefficients remain lower than for the other two learning algorithms, indicating less tight communities. Yet we note that after the second round of queries the average clustering coefficient increases around 75% and 25% in the in-topic and off-topic scenarios, respectively, and stabilizes at these levels in the long term. This shows that peers have the capability of forming and maintaining communities even with a very simple learning algorithm.

<sup>2</sup><http://lucene.apache.org/nutch/>



**Figure 1: Average change in clustering coefficient and diameter as a function of the number of queries sent by each peer for random-known, greedy and reinforcement learning algorithms. The two plots correspond to the in-topic (top) and off-topic (bottom) scenarios.**

We see from Figure 1 that peers become more tightly clustered with the more sophisticated learning algorithms. In the greedy algorithm the term weights in peer profiles can only increase. This leads to a more stable connectivity pattern and a higher clustering. In the reinforcement learning algorithm a neighbor can be penalized, leading to less stable patterns and a decrease in clustering after the fifth query. The particularly high peak in clustering observed for the greedy algorithm in the off-topic scenario is an artifact of the deterministic way in which ties are broken. A randomized update sequence would eliminate the bias in favor of peers discovered early and eliminate this spurious effect.

## 5.2 Semantic Locality

To illustrate the emergence of semantic locality in the network we clustered the peers using a community discovery algorithm, namely a fast variant of the top-down hierarchical clustering method proposed by Girvan and Newman [10, 24]. Figure 2 illustrates a few of the emerging communities using the greedy algorithm after each peer has processed 50 queries. We note the semantic locality of the clusters, with peers clearly routing queries to topically similar neighbors. To quantify this notion we next analyze the quality of each peer’s neighbors via topical semantic similarity measurements.

Figure 3 shows the average semantic similarity gauging the quality of each peer’s neighbors from our simulations. All three learning algorithms start with the same similarity value, but after five queries issued we observe differences in performance among the query routing schemes. The adaptive query routing schemes take advantage of learning and improve their performances over time.



**Figure 2: Dendrogram of representative neighbor clusters after 50 queries with the greedy learning algorithm. Each peer is labeled by its ODP topic.**

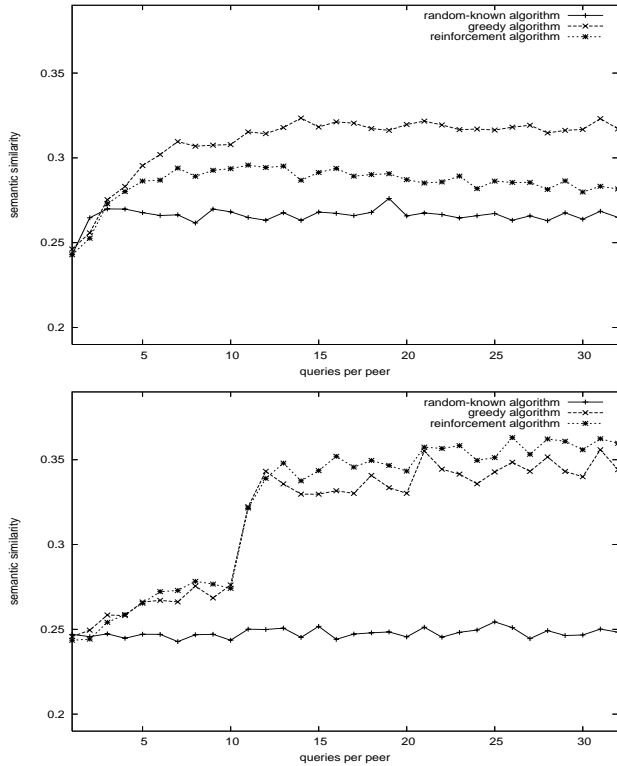
The greedy and reinforcement learning algorithms obviously outperform the random-known algorithm in both query scenarios; the latter fares slightly better in the in-topic case.

We see from Figure 3 that the reinforcement learning algorithm outperforms the greedy algorithm for off-topic queries while the reverse is true for in-topic queries. The reason lies in the different peer profile update rules. A reinforcement learning peer  $p$  does not change the profile weights of neighbors who do not return any response for query  $q$ , but it penalizes neighbors returning hits that are not as good as  $p$ ’s local hits. On the contrary, in the greedy algorithm a peer does not penalize neighbors who return low-quality results. In the in-topic situation a peer supposedly knows best about its queries since they are about its own topic. The rest of the peers are unlikely to provide better results, so they are penalized by the reinforcement learning algorithm even when they are the best targets for the query. The greedy algorithm is able to identify appropriate neighbors even if they provide worse results than local ones. This explains the greedy algorithm’s better performance. In the off-topic case, the reinforcement algorithm can identify the best neighbors because they provide better results than the local ones. This yields the highest semantic locality in Figure 3.

Combining these semantic locality results with the network topology evaluation in Figure 1, we now understand that the clusters formed by the greedy and reinforcement learning algorithms after ten queries correspond to the discovery of semantic locality in the peer network. In contrast, the random-known algorithm allows peers to cluster, but neighbors are not semantically related. The semantic locality measurements in Figure 3 also confirm that the off-topic clustering peaks of the greedy and reinforcement learning algorithms during the first ten queries are indeed artifacts.

## 5.3 Quality of The Search Results

Having shown that semantic communities can emerge in 6S, we need to test our hypothesis that these communities lead to better performance. But before applying the global coherence and coverage measures to this end, we conducted a preliminary experiment to verify that they can be used for such performance evaluation. In this



**Figure 3: Average semantic similarity between each peer and its neighbors for the three adaptive query routing algorithms with in-topic (top) and off-topic (bottom) queries.**

experiment we compared the two measures on two sets of URLs: *relevant retrieved* (i.e., URLs classified in the topic under consideration) and *random retrieved* URLs selected uniformly from the entire ODP. We expected the measures to return higher values for the relevant set than for the random one.

A performance evaluation based on our criteria requires the evaluator to collect a (not necessary complete) set of relevant pages. From these, a set of terms are taken to characterize potentially relevant resources (a relevant set  $R$ ) for a given query. For our task we used the ODP directory to construct relevant sets as follows. Let  $t_1, \dots, t_m$  be third level topics in the ODP directory and let  $q_1, \dots, q_m$  be  $m$  queries associated with these topics. To construct a relevant set  $R_i$  for each query  $q_i$ , we extract the descriptions of URLs from the ODP subtrees rooted at the topic  $t_i$ . Each  $r \in R_i$  is then defined as a set of keywords extracted from these descriptions and it represents a potentially relevant result for query  $q_i$ .

In the preliminary experiment, we used  $m = 50$  ODP third level topics and applied the procedure described above to construct the relevant set  $R$ . For a given topic, the relevant retrieved set ( $A_{relevant}$ ) was created using 10 URLs within that topic subtree in the ODP. To construct the random retrieved set ( $A_{random}$ ), we used a similar method but, instead of extracting URLs from the subtree under the relevant topic, we randomly selected 10 URLs from the whole ODP directory. Finally, we validated global coherence and coverage by comparing their values on the two retrieved sets. The results (omitted for space limitations) show a statistically significant improvement in global coherence and coverage from the random to the relevant retrieved sets. This confirms that global coherence and coverage are feasible for measuring the performance of a peer search system.

Let us now apply this evaluation approach for assessing the per-

formance of the peer-based Web search system. To this end, we considered the top 10 hits for each query  $q_i$  retrieved by the peer-based search system in our simulations as the retrieved set ( $A_i$ ). To construct the relevant set ( $R_i$ ) for each query  $q_i$ , we used the same method as the one used for the above preliminary experiment.

Figure 4 shows that the quality of the results returned by the greedy and reinforcement learning algorithms are significantly better than the random-known algorithm both in terms of global coherence and coverage. But there is no significant difference in the quality of the results returned by the greedy and reinforcement algorithms. Additionally, the quality of the search results for the three query routing algorithms increases with the number of query-response interactions.

Note that our earliest measurements of global coherence and coverage are taken after the first query and its responses have already propagated through the network. This explains the different performance between random-known and the other two algorithms.

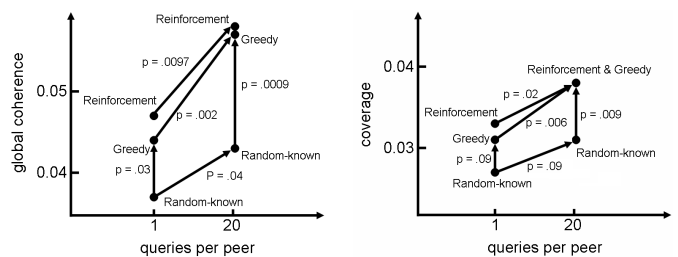
## 6. DISCUSSION

The experiment results show that 6S peers can learn from their interactions to form semantic communities even when the network is unstructured. This leads us to believe that it is possible to create a system which can enjoy the benefit of semantic communities without a structured overlay network to impose semantic similarity. We also find that the combined measurements of network topology and semantic locality can be used to gauge the efficiency and appropriateness of the network's emergent communities. With the global coherence and coverage measures, we show that the formation of semantic communities corresponds to an increase in the quality of the search results.

Our evaluation of semantic communities has many directions for further development. One important future task is to apply our evaluation to other types of peer-based search applications. With respect to the 6S system, it is desirable to study methods for merging results from peers, which may use different ranking schemes, and analyze the robustness of our findings about the topology when peers enter or leave the network.

A combination function for topical semantic similarity needs to be developed for the multi-topic problem. In our simulations, users and queries are only associated with one topic, but this is not a general requirement for a peer-based search application. In reality a user may have more than one topic of interest and a query may fit into several different topics. We are working on the extension of the proposed measures to the multi-topic situation to combine different topics into a final assessment.

If one wanted to study the emergence of semantic communities in a peer network with real rather than simulated users, a topic classifier would need to be developed for assigning topics to users and



**Figure 4: Average global coherence (left) and coverage (right) of random-known, greedy and reinforcement learning algorithms. We draw edges between two algorithms only when the  $p$  value of a t-test comparing them is less than 0.1.**

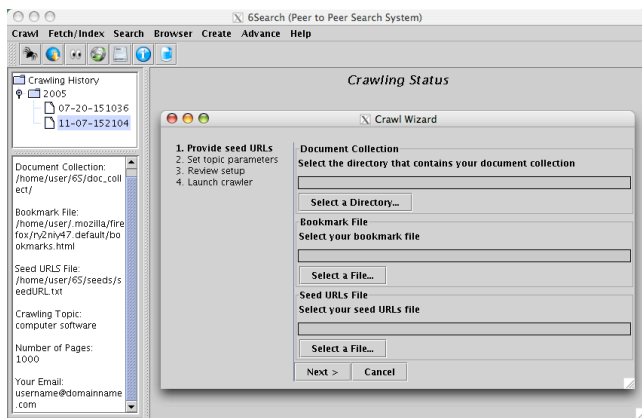


Figure 5: Screen shot of the 6S application.

queries. Our topical semantic similarity calculation is based on the semantic similarity between any two nodes in the ODP taxonomy. In the simulations presented here, peers and queries are designed to be associated with OPD topics, but in the real world a user may not know how to characterize the topics of her peer indexing system.

Windows, OSX and Linux prototype versions of the 6S application are available for alpha-testing.<sup>3</sup> Figure 5 offers a view of the user interface. The prototype, based on the JXTA framework [32], integrates the 6S protocol, topical crawler, document index system, search engine system and network communication system. Testing the prototype in a realistic setting will help us to study the robustness of the system from a security standpoint, e.g., with respect to denial of service attacks. Testing the prototype “in the wild” will also allow us to tune our protocols and algorithms. For example, while a peer may decide not to share its knowledge with other peers, we will consider whether the information available to a peer should be dependent on what it is willing to share.

## Acknowledgments

We are grateful to the Nutch Organization for its open source search engine code and to the Open Directory Project for the data used to model our simulated users. Work supported in part by NSF Career Grant IIS-0348940 to FM.

## 7. REFERENCES

- [1] R. Akavipat, L.-S. Wu, and F. Menczer. Small world peer networks in distributed Web search. In *Alt. Track Papers and Posters Proc. 13th International World Wide Web Conference*, pages 396–397, 2004.
- [2] E. Amitay, D. Carmel, R. Lempel, and A. Soffer. Scaling ir-system evaluation using term relevance sets. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–17, New York, NY, USA, 2004. ACM Press.
- [3] S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Comput. Surv.*, 36(4):335–371, 2004.
- [4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, New York, 1999.
- [5] M. Bawa, R. Bayardo Jr, S. Rajagopalan, and E. Shekita. Make it fresh, make it quick — searching a network of personal webservers. In *Proc. 12th International World Wide Web Conference*, 2003.
- [6] H. Chu and M. Rosenthal. Search engines for the World Wide Web: A comparative study and evaluation methodology. In *Annual Conference Proceedings (ASIS'96)*, pages 127–135, October 1996.
- [7] A. Clauset and C. Moore. How do networks become navigable? Technical report, arXiv.org:cond-mat/0309415, 2004.
- [8] A. Crespo and H. Garcia-Molina. Semantic overlay networks for P2P systems. Technical report, Computer Science Department, Stanford University, 2002.
- [9] S. Deerwester, S. Dumais, F. GW, T. Landauer, and R. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.
- [10] M. Girvan and M. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 99:8271–8276, 2002.
- [11] S. Joseph. Neurogrid: Semantically routing queries in Peer-to-Peer networks. In *Proc. Intl. Workshop on Peer-to-Peer Computing*, 2002.
- [12] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti. A local search mechanism for peer-to-peer networks. In *Proc. 11th Intl. Conf. on Information and Knowledge Management (CIKM)*, 2002.
- [13] M. Khambatti, K. Ryu, and P. Dasgupta. Efficient discovery of implicitly formed P2P communities. *International Journal of Parallel and Distributed Systems and Networks*, 5(4), 2002.
- [14] I. A. Klampanos, V. Poznański, J. M. Jose, and P. Dickman. A suite of testbeds for the realistic evaluation of peer-to-peer information retrieval systems. *Lecture Notes in Computer Science*, 3408:38–51, 2005.
- [15] D. Leake, A. Maguitman, and T. Reichherzer. Exploiting rich context: An incremental approach to context-based web search. In *International and Interdisciplinary Conference on Modeling and Using Context, CONTEXT'05*, pages 254–267, Paris, France, July 2005. Springer.
- [16] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann Publishers Inc., 1998.
- [17] J. Lu and J. Callan. Content-based retrieval in hybrid peer-to-peer networks. In *Proc. 12th Intl. Conf. on Information and Knowledge Management (CIKM'03)*, 2003.
- [18] J. Lu and J. Callan. Federated search of text digital libraries in hierarchical peer-to-peer networks. In *Proc. 27th European Conference on Information Retrieval (ECIR)*, 2005.
- [19] A. G. Maguitman, F. Menczer, F. Erdinc, H. Roinestad, and A. Vespignani. Algorithmic computation and approximation of semantic similarity. *World Wide Web*, 2006. Published online at <http://dx.doi.org/10.1007/s11280-006-8562-2>.
- [20] A. G. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani. Algorithmic detection of semantic similarity. In *WWW '05: Proceedings of the 14th International Conference on World Wide Web*, pages 107–116, New York, NY, USA, 2005. ACM Press.
- [21] A. G. Maguitman, F. Menczer, H. Roinestad, and A. Vespignani. Algorithmic detection of semantic similarity. In *Proc. 14th International World Wide Web Conference*, pages 107–116, 2005.
- [22] G. Pant, P. Srinivasan, and F. Menczer. Crawling the Web. In M. Levene and A. Poulouvasilis, editors, *Web Dynamics*. Springer, 2004.
- [23] J. Pujol, R. Sangüesa, and J. Bermúdez. Porqpine: A distributed and collaborative search engine. In *Proc. 12th Intl. World Wide Web Conference*, 2003.
- [24] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proc. Nat. Acad. Sci. USA*, 101(9):2658–2663, 2004.
- [25] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pages 448–453, 1995.
- [26] T. Saracevic. Evaluation of evaluation in information retrieval. In *Proceedings of SIGIR*, pages 138–146, 1995.
- [27] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *Proc. INFOCOM Conference*, 2004.
- [28] C. Suel, J.-W. Wu, J. Zhang, A. Delis, M. Kharrazi, X. Long, and K. Shanmugasundaram. ODISSE: A Peer-to-Peer architecture for scalable Web search and information retrieval. In *International Workshop on the Web and Databases (WebDB)*, 2003.
- [29] C. Tang, Z. Xu, and S. Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. In *Proc. ACM SIGCOMM '03*, 2003.
- [30] D. Tsoumakos and N. Roussopoulos. Adaptive probabilistic search for peer-to-peer networks. In *Proc. 3rd International Conference on Peer-to-Peer Computing (P2P)*, 2003.
- [31] S. Voulgaris, A. Keramarrec, L. Massoulié, and M. van Steen. Exploiting semantic proximity in peer-to-peer content searching. In *Proc. 10th Intl. Workshop on Future Trends in Distributed Computing Systems (FTDCS)*, 2004.
- [32] S. Waterhouse. JXTA Search: Distributed search for distributed networks. Technical report, Sun Microsystems Inc., 2001.
- [33] D. Watts and S. Strogatz. Collective dynamics of “small-world” networks. *Nature*, 393:440–442, 1998.
- [34] L. Wishard. Precision among Internet search engines: An earth sciences case study. *Issues in Science and Technology Librarianship*, Spring 1998.
- [35] L.-S. Wu, R. Akavipat, and F. Menczer. 6S: Distributing crawling and searching across Web peers. In *Proceedings of the IASTED International Conference on Web technologies, Applications, and Services*, Calgary, Canada, July 2005.

<sup>3</sup>[homer.informatics.indiana.edu/~nan/6S/](http://homer.informatics.indiana.edu/~nan/6S/)