Sixearch.org 2.0
Peer Application for Collaborative Web Search

Namrata Lele, Le-Shin Wu, Ruj Akavipat, Filippo Menczer
School of Informatics
Indiana University
{nlele, lewu, rakavipa, fil}@indiana.edu

ABSTRACT
Sixearch.org is a peer application for social, distributed, adaptive Web search, which integrates the Sixearch.org protocol, a topical crawler, a document indexing system, a retrieval engine, a P2P network communication system, and a contextual learning system. With a single click, the Sixearch.org application will build your personal Web collection. You can search not only your collection, but also other Sixearch peers. When you submit a query, your Sixearch agent will determine which peers are best suited to answer it based on previous interactions. Your agent will also learn from the results it receives, so that it can continuously improve.

Categories and Subject Descriptors
H.4 [Information Systems Applications]: Miscellaneous

Keywords
Peer collaborative search, adaptive query routing

1. OBJECTIVES
The goal of Sixearch.org is to provide an open-source platform for developing a context aware personalized peer-to-peer (P2P) distributed information retrieval system. The application currently supports collaborative Web search with scalability.

To simplify development and integration of different functionalities and their corresponding user interfaces, the application is designed in a modular fashion with a Web based interface. We demonstrate modularity of the code base by creating two local search engine modules, one using Nutch (nutch.org) and the other using Google Custom Search Engine (google.com/coop/cse).

While traditional search engines such as Google and Yahoo provide access to very large document collections, the Sixearch.org application provides a complementary way for users to actively and collaboratively share their own document collections. However, the Sixearch.org framework allows traditional search engines to naturally be included as peers; such peers quickly emerge as reliable and general authority nodes [2].

2. THE MACHINERY OF SIXEARCH.ORG
Sixearch.org uses a distributed, collaborative approach to Web search but without assuming the presence of special directory hubs. As shown in Fig. 1, each peer is a content provider; it employs a topical crawler guided by its user’s information content, which supports a local search engine—typically but not necessarily a small one. In the current implementation, Sixearch.org relies on three open-source platforms: Nutch for one of the search engine modules, JXTA (jxta.org) for the P2P network communication, and Berkeley DB (oracle.com/database/berkeley-db) for the underlying database system. A detailed description of Sixearch.org’s protocols and algorithms is out of the scope of this demo and can be found elsewhere [3].

Briefly, each Sixearch peer learns and stores profiles of other peers with a view to their potential for answering prospective queries. A user’s query is first matched against the local engine, and then, by checking profile information, routed to neighbor peers to obtain more results. The Sixearch peer will also learn from the results it receives, so that it can continuously improve the probability of choosing the appropriate neighbors to route future queries. The peer network leads to the emergence of a clustered topology by intelligent collaboration between the peers [1].

3. FEATURES
The Sixearch.org application is implemented in Java to be a cross-platform system. It is designed to make it easy and transparent to use and manage the application without leaving a Web browser. From the user’s perspective, the main features of Sixearch.org are peer search, personal Web index management (auto and manual), and a Firefox extension. The user can...
employ a wizard to create a personalized search engine and then use a Firefox extension to automatically route the query to the network from the browser's search box.

**Peer Search:** When a user submits a query through the Web interface (Fig. 2), in addition to getting results from the local engine, results from other peers are shown and updated in real time as they arrive. The results and queries are also recorded (but not shared) to help the user find past useful results.

**Personal Web Index Management:** The application can automatically create a Web index by synchronizing with the user's bookmark file or importing from other resources, such as del.icio.us and givealink.org (Fig. 3.a). Users can also manually create or add to the personal index (Fig. 3.b) by running a topical crawler guided by a provided topic. The crawling results will then be indexed for keyword searching. At any time, the user can view, add, remove or tag any indexed document (Fig. 4). In addition to Nutch, Google’s on-line custom search engine can be used as a local search engine.

**Firefox Extension:** The Firefox extension allows tighter integration with the Firefox browser. To search, a user can select the Sixearch option from the search box of the browser and type in a query (Fig 5.a). In addition, the user can quickly access the application’s management Web page or view a live snapshot of the Sixearch.org query network by clicking on the Sixearch logo at the bottom right of the browser (Fig 5.b). To use the Firefox extension, the Sixearch application must be running as a background process.

4. **FUTURE DIRECTIONS**

There are many directions for future research. Two of our most immediate interests are trust management and adopting folksonomy concepts into Sixearch.org.

When Sixearch.org gains more users, it is inevitable that spammers will try to exploit the system to increase traffic to their spam sites. To avoid forwarding queries to spamming peers as well as colluders, intelligent trust management is required. Currently we are developing a promising trust management system, which leverages users’ feedback and other information gathered from peers’ interactions, for the next release of the application.

By switching to folksonomy concepts, peers will store and exchange information in the form of triples (peer, url, tag). Instead of relying on page content, new techniques from folksonomy research will be integrated to provide a community-driven way for the application to find relevant results.

5. **REFERENCES**

