Haystack: Per-User Information Evironments
MIT9904-08

Progress Report: January 1, 2000—June 30, 2000

David R. Karger and Lynn Andrea Stein

Project Overview

The Haystack project is investigating the ways in which electronic infrastructure can be used to triangulate among the different knowledges of an individual, of collegial communities to which we belong, and of the world at large. Its infrastructure consists of a community of independent, interacting information repositories, each customized to its individual user. It provides automated data gathering (through active observation of user activity), customized information collection, and adaptation to individual query needs. It also facilities inter-haystack collaboration, exposing (public subsets of) the tremendous wealth of individual knowledge that each of us currently has locked up in our personal information spaces. The Haystack-NTT project involves augmenting its customization, learning and adaptation, and inter-haystack communication.

Progress Through June 2000

In the first six months of this grant, we rewrote the core of our Haystack system to create a privileged kernel. that assures the persistence and transaction-safety of the data in a Haystack. We also began to investigate query self-organization of an individual's haystack in response to queries.

In the past six months, we have completed this work and in addition built on this functionality by providing rudimentary learning capabilities for haystack, by adding facilities to combine both structured and unstructured queries, and by integrating additional input facilities. Finally, Haystack can now archive substantial corpora (5K documents).

The Haystack data model is semi-structured; it allows both arbitrary text and structured relations such as authorship. During the past six months, we developed a system that performs database-like queries on semistructured data using current relational database technologies. To achieve this goal, we designed a database schema that would allow us to store our data model. We also specified the format in which the user can enter database queries and implemented procedures that translate these user queries into SQL. Finally, we designed and integrated these structural searches with standard text search within Haystack.

In a separate project, we added several machine learning techniques to Haystack. Now, when a user makes a query on a topic similar to a previous query, the system uses any relevance feedback available from that prior query to provide an improved result set for the current query. The core learning module was designed to be modular and extensible so that more sophisticated learning algorithms and techniques can be easily implemented in the future. Testing of our system shows that learning based on relevance feedback even using the simplest algorithm provides some improvement on the results of the queries.

Haystack is a ubiquitous information repository; it aims to collect all information with which a user interacts. In a final project over the past six months, we integrated scanning and OCR capabilities into Haystack. This extends the reach of Haystack's suite of information management tools beyond the purely electronic realm and into the paper office. The scan-to-haystack pipeline is fully automated; dropping a document into the scanner's feeder results in archival of the document's content.

Research Plan for the Next Six Months

Over the next six months, we will work on a variety of back end enhancements and improvements to increase the robustness of the system; we will begin construction of a new front end designed to allow increased user customizability; and we will continue to improve the performance of Haystack's archive and query functions to increase its utility over increasingly larger corpora.

On the back end, we will begin work on a more robust transaction manager to increase the system's fault tolerance and allow for backing out of erroneously derived data. We will also start to design a versioning system that will make it possible to query previous system states. Finally, we will undertake a re-modularization of the kernel system to allow increased extensibility as well as further evolution of the system.

We also plan to build a more object-oriented user interface that will display each object in the way best suited for that object. We intend to explore a new user interface for haystack based on hints. We expect to record a user's preferences regarding the display of a particular object – e.g., arrangement of individual objects within a folder – as Haystack data for use by multiple display tools as well as for potential use by some of Haystack's retrieval services.

By the end of December 2000, we expect to be able to demonstrate a robust Haystack system running on a sizable corpus exploiting several of these new features.