Haystack: Per-User Information Environments
MIT 9904-08

Proposal for 1999-2000 Funding

David Karger and Lynn Andrea Stein

Project Overview

The World-Wide Web will soon be a powerful and dynamic replacement for today's libraries and museums—public repositories of communal knowledge. But is this enough? Is this all that the electronic infrastructure can be? Libraries are huge, filled with masses of data irrelevant to any query. They are impersonal, presenting every user with the same information regardless of their background and interests. For these reasons, libraries are typically the last place we turn in seeking information. We start closer to home, with our own bookshelves. These are hand-selected and custom-organized collections of the information most relevant to us as individuals. If we cannot find what we seek on our own bookshelves, we turn next to colleagues, friends, and associates---trusted members of communities with interests common to ours. It is only after exhausting these resources that we go to the library.

The Haystack project is investigating the ways in which electronic infrastructure can be used to triangulate among these 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 representations, and adaptation to individual query needs. It also facilitates 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.

We currently have a very basic prototype of Haystack. The proposed work would involve augmenting its database system, customization, desktop integration, learning and adaptation, and inter-haystack communication. We would also consider exploring trans-lingual and multimedia applications.

Approach

The Haystack Project aims to create a community of individual but interacting "haystacks": desktop-integrated personal information repositories which archive not only base content but also user-specific meta-information, enabling them to adapt to the particular needs of their users.

- Every information transaction will be brokered by a user's haystack. All information manipulated by a user (files, mail messages, web pages) will become part of his personal collection without user overhead. Information in a user's haystack is accessible to the user at any time through augmented facilities integrated into normal desktop tools (such as an editor or a web browser). A user's haystack will be wholly integrated into his desktop. For example, a haystack user could "find file" in an editor using keyword search on remembered characteristics of the document rather than a precise description of the document's location in the file system.
- Unlike traditional information retrieval systems, Haystack will customize itself to each user. Haystack adapts over time to the behavior and vocabulary of its user, modifying its retrieval algorithms to reflect improved understanding of how its user describes information and information needs, and what kind of documents are best for addressing the user's needs. Part of this involves recording explicit preferences of the user. Much more involves behind-the-scenes work: metaphorically looking over the user's shoulder, collecting information on the user's behalf and watching what he does in order to adapt future haystack behavior.
- Different Haystacks will collaborate with each other, learning over time which users have related interests and helping to form "referral systems" that take a user's question to other haystacks that can answer it. Any improvement in information quality or organization achieved by one individual can then be used to advantage by all other participants. Recalling our bookshelf metaphor, we aim to provide tool that allow for the identification and interrogation of "kindred spirits" who collect and organize information an individual will find useful. Haystack customization will form a lens through which they individuals view and search for information located in others' collections.
- In addition, each Haystack has a pro-active "archiver" process that actively seeks to keep its collection up-to-date, adding information that might be useful to the user in the future.

By integrating adaptive, collaborative information retrieval into every aspect of a user's interactions with a computer system, Haystack aims to become the quintessential information assistant.

Why Haystack?

The motivating idea of the Haystack project is the distinction between libraries and office bookshelves. Typical information retrieval systems assume a large corpus which is queried by numerous anonymous clients. While this is clearly an important model, we contend that it is by no means the only one worth studying. The old-fashioned analogue to querying such a system is a trip to the library: a repository of massive amounts of information that probably contains (somewhere) what we are looking for.

In practice, the library is the last stop on a person's hunt for information. When a person looks for information, he will often start with his own bookshelf. This "personal repository" contains a collection of information, built up over time, that reflects the needs and knowledge of its owner. This makes it different, in crucial ways, from the library. For

example, all the content was actively placed there by the user, who is familiar with it and believes it to be useful. In the user's area of expertise, it is often more up to date than the library. Overall, a person's bookshelf contains the bulk of the information that he considers most valuable.

An individual's bookshelf is also organized in an idiosyncratic fashion. While library materials are arranged according to a standardized classification scheme, individuals have been known to arrange their books by topic, chronology, usage pattern, or even size and color. Even users who make no active attempt to organize their books find them structured in some kind of most-recently-used hierarchy. Individuals exploit their idiosyncratic organization when searching for information: they may look for a blue book, or a book on the bottom shelf, or a book next to another book. At a library, users are limited to searching the standard classification.

If a person's bookshelf fails him, he still has an alternative to the library. The natural next step is to ask his colleagues in neighboring offices. Turning to a colleague offers several advantages over a trip to the library. Colleagues have their own personalized collections of information which they can search effectively. They share interests and vocabulary with the original questioner, and are thus likely to be able to understand that person's information needs and effectively communicate anything they might know that can help. A person can describe his problem in a language common to him and his colleague, and his colleague can then use her own knowledge of her collection to find what the original searcher wants. Finally, books in colleagues' personal collections are more "trustworthy" than random books selected from a library. Their presence in the colleague's collection indicates that someone we trust considers them valuable.

Specifically, the Haystack project is motivated by (and aims to answer) the following questions:

1) Current IR systems have a "one size fits all" flavor. Alta Vista gives you the same search results to a query whether you are 5 years old or 50. If we take the varying backgrounds, skills, habits and preferences of the individual user into account, can we devise a system that delivers better performance to them?
2) In order to deliver individualized performance, we need to learn about the user. How can we do so without interfering unduly in the user's activities? In particular, what kind of information can we learn from passive observation of the user? One obvious informative element is what the user has stored locally (which suggests they consider it valuable). What use can be made of all this information?
3) Each individual devotes some time to evaluating and organizing the information they encounter. Can we disseminate those bits of evaluation and organization to other people, so that they can make use of them too?

Specific Steps and Strategies

In order to make Haystack work, we need to address several issues described here.

The data model

One important way that the Haystack project differs from others is in the kind of information it organizes. While typical IR systems focus on the indexing of text (or close variants such as html), Haystack attempts to archive everything of interest to its user. In particular, we aim to archive both the content and context of every document. Context includes metadata about document, for example, includes its location, its history, its authorship, its relevance to past queries, when it was last accessed, the desktop application that supplied the data to Haystack or the time and date of archiving. It may also include specialized type-dependent information about the object, such as the sender and recipient of an email message. In Haystack, this meta-data is first class: it is itself information to be archived and retrieved. This way we exploit not only the text of the document but also its organization and its relationships to other documents already in the collection. All of these aspects are available to subsequent queries, so that it is possible to ask for “the article I sent to Chris last Thursday in response to his email message”.

In order to represent all this information, we use a very generic data model of “straws” connected by named “ties”---for example, a document straw may be connected to a straw containing a person’s name via an “author” tie. The user can add straw and tie types as he sees fit to represent whatever data and relationships he finds useful.

We do not expect the user to gather all this data. Instead, we provide an extensible family of independent agents that opportunistically extract data from objects in the haystack: for example, an agent might be able to identify the title of an html document by understanding the format of such documents.

The Interface:

A key goal in haystack is to gather as much information as possible from and about the user---data that can be used to tune the system to its owner One source of data is active annotation by the user. Our interface lets a user easily add to or correct the information that is automatically extracted into the haystack. . In many cases, the user knows better than the system. We give them easy tools for adding to or overriding the system generated metadata. We let the users jot “notes in the margin”---comments on the data.

But active annotation by the user is an undesirable drain on their time. Thus we emphasize passive observation of the user. For example, we provide web and email proxies that observe the browsing and mailing activities of the user., recording into the haystack all web pages a user browses and all mail the user sends. We also carefully observe user queries to the haystack, in order to see how the user refines queries, and what retrieved objects the user actually visits (indicating they were what he wanted). Eventually, Haystack could be integrated into the shell (or file system) itself. Some of this information is the kind user would like to actively exploit (“I want the mail I sent to Lynn last week”). Other information can tell the system about connections between objects (if two documents were edited at the same time, they are probably related). The benefit of the passive approach is that it gathers data {\em without} adding to a user’s

workload. Integration into the user's desktop decreases the overhead of using haystack to the point that we believe most users will interact with a significant portion of their data through our system.

Learning

With the data gathered into the haystack, we aim to adapt the system to make it more effective for its user. Techniques from machine learning can come into play here. As an example, consider learning from a query. Our interface can observe a user typing a query and also note what objects in the result set the user actually chooses to look at. Obviously, if the user types the query again at a future data, we can show him the objects he chose to look at the first time. Using machine learning, we can also *generalize*, using the users previous choice to improve retrieval performance on *different* queries.

Several tools from the learning community can be applied effectively to this problem. There has already been a great deal of study of how learning algorithms can be applied to information retrieval systems. In particular, learning algorithms have often been used to train information filtering systems that select "interesting" documents from a stream of newly arriving documents. While this ignores the issue of queries seeking documents on a particular topic, we believe such tools can be adapted to train a system on how to answer queries based on observed reactions to previous answers to similar queries.

Collaboration

We can amplify all of the benefits described above if we allow individual haystacks to interact. The annotations and retrieval decisions that one user makes can be of great value to other users (particularly colleagues with similar tastes and vocabularies). The mere fact that a given document is in many people's haystacks, even unannotated, suggests that it is a document of high intrinsic value---we might prefer it over documents that are apparently better matches to the query but are less popular. We aim to export information about which documents are in which user's haystacks, what annotations users have made to the documents, and what documents users have selected as being relevant to particular queries. Both to limit the costs of a search and to improve the filtering of what is returned, it is important for the system to learn over time which other individuals are most likely to have information that a given user finds relevant---these haystack "neighbors" are the systems that should be queried first and whose results should be most trusted.

The combination of personalization and collaboration creates a useful alignment between individual's supplies of and demands for information. Past work on collaborative filtering (see below) has expected individuals to behave altruistically, spending time ranking documents after they have found them, when they will no longer accrue benefit from doing so. Haystack, in contrast, asks people to perform only the information organization activities that will help {\em them}, and then exports the outcome of those activities to others. Haystack fits the modern economic model of individuals behaving selfishly and this contributing to the general good.

We also feel that our haystack approach will encourage even more dissemination of information than is presently undertaken. Right now, there is overhead in publishing one's information: the publisher must take active steps to present it in such a way that it is accessible. This requires that the publisher properly organize the information to allow others to find it. With Haystack available to take care of the organization, publishing a document will be as easy as deciding what to make public.

Another opportunity that the linking of haystacks creates is in connecting individuals to other people who can address their information need. The information I have stored in my haystack is likely a good indicator of my knowledge and interests. A question that matches a lot of material in my haystack is likely to be a question I can usefully answer. The haystack system can therefore serve as an "information brokerage" connecting questioners to experts.

Status

A prototype of the Haystack system is currently installed on several MIT networks. This version of the system supports indexing of numerous data types, annotation by users to reflect their own interests, and standard text searching capabilities that let users retrieve information. It exhibits some basic adaptability to its users; for example, if a user performs a search and ends up using a particular document, the system "binds" that search to the document so that future searches like it will preferentially retrieve that document. We are currently at work stabilizing and scaling the system to deal with reasonable quantities of information.

Deliverables by end 1999

Assuming we succeed in funding graduate students to work on the project,we expect to demonstrate a quite useful system by the end of 1999. The system will scale to handle the tens of thousands of documents or mroea typical user owns. It will be able to handle many types of data and will posess agents that extract a great deal of useful information. We will demonstrate structured search, browsing, and personalized annotation of a collection of archived documents. We may have begun on the machine learning components of the system, but are unlikely to be finished. The collaborative elements of the system will be built later.