Haystack: A Personalized IR System

David R. Karger and Lynn A. Stein
MIT

A great deal of research in information retrieval has been detached from the users who could eventually benefit from such research. On the one hand, traditional IR research systems have been cut off by an inconvenient interface or limited to an unchanging text collection with a fixed set of evaluation queries that become increasingly out of date as time passes. On the other hand, the recent surge in web search tools has resulted in many deployed IR systems, with more convenient but often limited interfaces and generally non-customizable corpora. A few systems, such as Harvest or Content Routing, have attempted to address the gaps between these two extremes, focusing on the construction of a more flexible substrate which allows users and communities to build their own repositories or queries.

The Haystack project is aimed at the individual customization end of these more realistic "living" information retrieval systems. We are interested in building on customizable substrates, such as those provided by Harvest or Content Routing, to create a community of individual but interacting "haystacks": 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. We believe that such a system will let us address several questions:

- How can individuals use an information retrieval system to organize their own personal collection of information?
- How might an information retrieval system learn from its users and evolve over time into a more effective system?
- As individuals build up their own collections and information retrieval systems, how can they search for information that might be located in others' collections, especially when such information is organized by information retrieval systems that may differ greatly from their own?

Our first step towards this goal has been to design a simple and convenient user interface to and annotation format for an information retrieval system. Our current annotations emphasize user-independent text meta-information, but the format for and structure of these annotations are intended to encompass hand-generated and automatic user-specific annotations. The annotations themselves are first-class documents in our system, so that, for example, search information can be reified and treated as an indexable object.

In our implementation, we have chosen to detach the information retrieval "engine" from the user interface and annotation system, specifying only that the engine should accept a natural language query and return documents that "match" under whatever criteria it uses. We have begun by using the "MG" information retrieval system, but are concurrently investigating other "back ends" including Content Routing, Harvest, and an in-house image-based IR system.

On top of this arbitrary engine, we are implementing several interfaces for retrieval as well as annotation editing. The first is a web-proxy based interface, which allows users to connect to the information retrieval engine via their favorite web browser. To maximize ease of use, we are also developing shell- and emacs-based tools for talking to the collection. Haystack is intended to archive any objects from which text can be extracted; we are initially implementing (or appropriating) "textifiers" for ascii, postscript, html, and scanned documents, but have an architecture that is easily extensible to other documents types.

It is our intent that the simple standalone version of Haystack will be easy to integrate into everyday use. Since we project that even a minimal system will be of use to people aiming to organize their mail, file system, and favorite web pages, we therefore expect to attract a moderate-size community of users at MIT. Once the system is in use, we will be able to leverage the annotation facilities to explore several questions.

The first such question is how an information retrieval system will actually be used in practice. By gathering usage data (with permission) we hope to learn about the kinds of queries people typically use. Are they usually boolean in nature? Single words? Do they tend to be over-precise and find no documents, or do they overgeneralize and get swamped with useless results? How do they react to what comes back? What refinement strategies do they use? Each haystack will provide a user-specific set of answers to these questions.

A second question is how a system might learn from interaction with its user. Consider the scenario in which a user types an initial query Q, then undertakes several stages of refinement to home in on the document D he wants. For the future, the system should learn that when the user types a query like Q, document D is likely to be relevant even if it does not appear to be a good match. The annotation system allows for both user and system-level support for this learning process. The system might annotate a document with terms that do not appear in it but that the user types when he expects to find that document. The user might also add keywords or mnemonic phrases to a document in the expectation of future searches for it. Ultimately, the system may be able to make user-specific generalizations based on automatically or manually entered "optimization" annotations.

Given that individuals are organizing the information they care about, it is natural to ask how one user can benefit from the work of other users. Consider that the typical way to search for a paper book is to ask one's office-neighbor for it. Analogously, we would like to let individuals search for information in other people's haystacks. 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.

Another opportunity that this 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.

Sharing haystacks also raises the issue of generalizing from individuals' customization of their own haystacks to larger (pooled) data-sets. This provides another opportunity to test the adaptability of query strategies and a test of the generalization of the underlying learning algorithms.

The common thread among the above ideas is user-specific customization of information, repositories, and retrieval processes. These are issues that are possible to explore only in the hybrid world provided by the newest generation of information access tools. By developing the Haystack system, we will attract the community of users who will provide the necessary testbed for exploring these questions about evolving, interacting customized information systems.