A Testbed for Proactive Information Retrieval

Miikka Miettinen, Ville H. Tuulos, and Petri Myllymäki

Complex Systems Computation Group
Helsinki Institute for Information Technology, Finland
firstname.lastname@hiit.fi

Abstract. In this paper we present our view of proactive information retrieval as a complementary interface to an index-based search engine. Relying on its global view of the available contents, the system would identify semantic relations between disparate sources of information, and provide the user with improved awareness of the available opportunities and alternatives. We outline functional requirements for the system, and propose some initial steps towards implementing it in practice. We describe our testbed, which enables us to evaluate the performance of alternative implementations instantaneously. We also compare and contrast our approach to some previous work.

1 Introduction

A proactive system takes actions on behalf of the user without being under explicit control [1]. In this paper we discuss the steps required to supplement an existing Internet search engine with capabilities for proactive retrieval. The idea is to observe the *navigation* and *scrolling* patterns of the user, and generate queries automatically to provide additional links to potentially relevant pages. At a later stage of our project we will consider using *eye movements* as a supplemental source of implicit feedback.

We think of keyword search and proactive search as two complementary interfaces to an underlying infrastructure. Keyword search is often the most efficient way of locating an initial set of relevant pages, and is needed for bootstrapping the proactive interface. On the other hand, proactive retrieval of additional links would encourage balanced exploration of the available contents and reduce the cognitive demands of the task, as we will explain in the next section.

The development of the proactive interface requires systematic experimentation. It is composed of several separate but closely interacting functions, including analysis of the user's actions, generation of the queries, and retrieval and ranking of the results. Careful investigation of the individual component problems is necessary for identifying opportunities for improvement and understanding how particular changes might affect the overall performance of the system. Furthermore, the effort should be guided by realistic data, as many important details depend on the specifics of the users' behavior.

The scale and scope of actual user testing that can be carried out in practice is very limited, however. Controlled experiments are too slow and laborious for comparing more than a few alternative implementations, and in less constrained settings the analysis of the results is difficult. We have tried to overcome these problems by constructing a testbed, which enables the majority of the experimentation to be done with a simulation. The idea is to rely on one reasonably realistic data set gathered in controlled circumstances, and test the proactive interface by looking at the number of relevant documents that it *would have retrieved* for the users when they were working on specific tasks. This kind of approach is sufficient for the purposes of development, but the utility of the end result will be assessed in a separate experiment.

In the next section we suggest some potential uses for proactive information retrieval and clarify its relationship to keyword search. Our approach to building the proactive interface on top of an existing search engine is discussed in section 3. The initial stages of the work will rely on the testbed, which is described in section 4. Section 5 compares our approach to some previous research, and section 6 concludes with general reflections on the issues involved.

2 Motivations for doing proactive information retrieval

Information retrival can usually be associated with a specific goal. According to [2], the variety of goals can be characterized in terms of a three-level hierarchy. At the highest level, a goal can be considered navigational, informational or resource-oriented by nature. A user with a *navigational goal* wants to visit a particular Web site, and does not care about alternative locations containing similar content. In contrast, an *informational goal* implies that the user wants to look up or learn something, and the source of the information is typically less important than the content. Informational goals can be divided further into a number of categories, of which the most relevant ones for the present discussion are directed and undirected goals. An informational goal is considered *directed* if the user has a particular perspective in mind, and wants either a single answer to a straightforward question or a gradually deepening answer to an open-ended one. In the case of an *undirected informational goal* the need is less specific, and the user is in principle interested in anything related to the topic. Finally, a user with a *resource goal* is concerned with acquiring something that is not considered information in the everyday sense of the word (e.g. a movie).

It is clear that with some of these goals *proactive* information retrieval cannot possibly work. In particular, narrow and specific goals are too difficult to recognize in sufficient detail for accurate retrieval to be possible. The recognition of a navigational goal, for example, would literally require reading the user's mind, and many informational goals are likely to be intractable also.

Let us assume that the user is looking for a single piece of information on the Web. People have been observed to avoid making complex queries to a search engine, and prefer navigating gradually towards the desired information instead [3]. In principle, this creates an opportunity for proactive retrieval. By analyzing the navigation path, the proactive system would locate the right piece of information before the user and provide a handy shortcut. This is unlikely to be feasible in practice, however, unless the navigation paths contain substantial amounts of repetition that can be harnessed effectively. The contents of the pages along the path almost certainly do not provide a basis for identifying the right fragment of text within a corpus of billions of documents, although with good luck it may of course occasionally be possible to “find a needle in a haystack”.

On the other hand, open-ended inquiry and undirected exploration could perhaps be supported in useful ways with proactive information retrieval. In these cases the user is reading systematically about a particular topic, and suggestions of other related pages would enable *opportunistic navigation* to promising directions. The contribution of the system would be based on its *global view* of the contents. At least in principle, the system could identify semantic relations between disparate sources of information, and provide the user with improved awareness of the available opportunities and alternatives.

In addition, a proactive system might reduce the cognitive demands of the task. In the absence of it, the users need to engage frequently in secondary activities associated with making search engine queries. *Text comprehension* is believed to require construction and continuous updating of multilayered mental representations, and additional cognitive load is likely to reduce performance [4]. The generation of effective queries, in particular, requires a specific viewpoint that differs from the primary task (see e.g. [5]). Frequent switches between tasks consume additional cognitive resources, and people have been found to avoid such interruptions when the situation is under their own control [6]. As a result, it is possible that keyword search is not used as effectively as it could be during activities involving extensive reading. Whether or not the situation could be improved with a proactive interface depends to a large extent on the accuracy of the results and the way they are presented to the user.

3 Elements of a proactive search engine

The feasibility of using a search engine in proactive information retrieval stems from the following two assumptions. First, we assume that the corpus of information to be searched is fairly static and known beforehand. Secondly, we assume that we can track certain implicit features of the user behavior which can be used to discriminate relevant and irrelevant pieces of information in the corpus. The first assumption postulates the system’s greatest benefit over the user’s mental model: It has a global, easily accessible view to the corpus. The second assumption is the system’s greatest challenge: How to infer cues of relevance given some noisy and implicit input. The actual task of designing a proactive search engine boils down to harnessing the former assumption to solve the latter.

Generally speaking, there are two closely related alternatives in forming the global view to the corpus. We could build a model of the corpus by making some relevance judgements *a priori*. For instance, we could try to capture some statistical invariances in the documents and cluster those documents together which seems to share similar features. According to the *cluster hypothesis* [10] of information retrieval, similar documents within the same cluster often share the same semantic characteristics. Another option would be to model some lexical invariances and model the general themes or topics appearing in the corpus [9]. This alternative relies on the assumption that the models are able to capture such features of the corpus which later on benefit the relevant document discrimination given the user's input. Models of this type are often lossy i.e. they deliberately lose some information.

The second alternative is index-based. We try to keep *a priori* assumptions in minimum but the whole corpus is indexed so that it will be easily and efficiently accessible in real-time. Both the alternatives are viable for a proactive search engine. However the index-based approach gives a greater weight to the user's input in contrast to the contents of the corpus. This makes it more suitable to noisy environments such as the Web where statistical distributions of the contents of documents don't always resemble interests of a user reliably. Furthermore the index-based approach allows us to experiment with different kinds of implicit input in a more flexible manner. Thus the second path was chosen.

We have identified the following features that are required from the search engine in a proactive setting like ours:

- **Coherent**. Even though the user doesn't have to understand the functional details of the system, the user must be able to form even a vague mental model on the behavior of the system. Especially this requires that the system behaves similarly in similar contexts.
- **Content based**. The suggestions by the system must be based on the fragments of text which have received the user's attention or which have been otherwise deemed relevant by the user.
- **Robust**. The system must not get confused by sporadic irregularities in the input.
- **Real-time**. The system must update the list of suggestions in real-time without noticeable lag.
- **Reactive**. We can be proactive only with respect to the history. If the user's needs change abruptly and the history becomes irrelevant, we must adapt to the new situation without unnecessary delay.

It is useful to consider a baseline system fulfilling these requirements. First, the last two goals, being **reactive** and **real-time**, may be achievable with a sliding window of history which follows the user's actions. Length of the transition period from one task to another, during which the system's suggestions may be mixed up, is proportional to size of the window. Thus we may control reactiveness by altering the window length. By keeping the window small, we may restrict the amount of query information and keep the response times in an acceptable scale.

The next two goals, namely being **robust** and **content-driven**, are mostly challenges of language modelling. The desiderata is the same as with any sophisticated information retrieval system and the methods developed therein are applicable.

We see that our greatest challenge is to keep the system transparent and **coherent** in the user's point of view. It's hard to overemphasize the importance of this goal, even though it is often shadowed by other more technical goals. We recognize the fact that any system for proactive information retrieval will suffer from various fallacies, due to complexity of the task. Yet we believe that the system does not have to be perfect in order to be usable. We must achieve an acceptable tradeoff between usability and known deficits.

This is a major motivation for carefully building a controlled testbed for the system. As we are fully aware of the corpus and relevance assignments with respect to various tasks, we may analyze the behavior of the system in detail. We hope that this knowledge enables us to tie the various modules of the system together in a coherent manner.

4 Description of the testbed

We need to make systematic comparisons of alternative approaches and implementations during development. Actual user tests cannot be carried out in the required scale, however, because of their prohibitive cost in terms of time and other resources. What we need instead is a *simulated test*, which gives us a rough indication of the performance of an individual implementation. Such tests involve generating queries on the basis of a specific data set, and using the number of relevant documents retrieved as the measure of performance. Our intention is to experiment in a somewhat simplified setting first, and move to larger and less structured portions of the Web as our understanding of the problem improves.

The construction of the testbed required real data to be gathered in controlled circumstances. The level of detail is sufficient for reconstructing the users' activities from the logs, and the relevance of any document appearing in the results can be determined automatically. The most straightforward way to meet these needs was to design an *experiment* based on a *restricted document collection* and *specific tasks*. In this kind of a setting we have precise knowledge of the relevance of each document, and are able to compute the performance measure instantaneously by feeding the available data to the proactive interface and looking at the results that it would have provided to the users.

We selected the documents manually from Wikipedia.[1] Compared to a general collection of Web pages, the resulting corpus has a number of desirable properties that make it suitable for us at this stage. The pages are freely available for downloading and adaptation to our special needs. The HTML is fairly clean and uniform, which eliminates a major source of practical difficulties associated with the creation of the search engine index and the monitoring of the scrolling patterns. Finally, the quality of the contents is surprisingly high within the domain chosen for the initial experiments.

For practical reasons, we used computer science students and researchers as subjects in the experiment. After extensive reading of Wikipedia and reflection on the needs, we identified *computer security* as the most appropriate topic for the tasks. It includes exciting and humorous elements that make it appealing to

[1] http://en.wikipedia.org

a wide audience, and the availability of high quality content turned out to be superior to most other topics. The subjects were likely to have some prior understanding to guide them during the course of the experiment, but they almost certainly would not have been able to perform the tasks adequately without the help of the material. This is typical of the kind of information retrieval activities that we are interested in, and contributes to the validity of the setting. Finally, many of the concepts of computer security are associated with distinctive vocabulary, which increases the probability that the goal we have set ourselves is achievable at least in principle.

The experiment was based on three broad tasks, each of which is associated with 14-19 different documents. The tasks were presented one at a time, and the subject was instructed to locate and study the relevant material as comprehensively as possible. The amount of time available for each task was 10 minutes, but the subject was allowed to move to the next task earlier if she felt that she had already covered all available material. The pages were accessed by means of *keyword search* and *hyperlinks*. At the beginning of a task, the subject located an initial set of potentially relevant pages with the search engine. Promising links and additional search terms led the subject further, until the time was up or everything seemed to be covered.

In order to get realistic scrolling data, we had to give an incentive to *read* the material in addition to visiting the pages. Therefore, we told the subjects that they would be asked questions about the contents at the end of the experiment. There was no need to actually do this, however, as the effectiveness of the users' cognitive strategies is an issue far beyond the scope of our research.

Basically, we were trying to simulate a situation in which the user is *gathering information* for a specific purpose. Typical examples of activities that include this kind of information retrieval are preparing a presentation or familiarizing oneself with a potentially interesting topic. The individual tasks in our experiment were related to three different issues concerning computer security:

1. The use of cryptography for improving the security of e-mail.
2. Security risks appearing in organizations because of the ignorance and carelessness of ordinary computer users.
3. Principles and practices facilitating the development of secure software.

The subjects were also given specific instructions regarding the relevance of particular types of content. For example, in the first task we stated explicitly that the inner workings of cryptographic algorithms are not within the scope of the task, but concepts and terminology are.

The corpus consists of 150 documents on *computer security* and another 150 documents on a variety of *other topics* related to information technology. The documents were selected from Wikipedia manually, and special attention was paid to their relationship to the tasks. In particular, an attempt was made to exclude any documents with unclear relevance. A corpus in which each document is either relevant or irrelevant to a specific task makes it easier for us to analyze the behavior of the system and increases the reliability of the results.

A unique characteristic of Wikipedia is the abundance of links. As our objective in the longer run is to work with large and heterogeneous collections of ordinary Web pages, it was appropriate to modify the link structure to make the corpus reminiscent of a miniscule version of the Web.

Some of the documents contained a "See also" section with links to other related pages. These were removed, because in general authors do not provide comprehensive listings of other available material. The internal table of contents appearing on some pages was removed to increase the opportunities for analyzing scrolling data. Finally, all links pointing outside of the corpus were removed, because the subjects were not allowed to use external resources during the experiment.

Wikipedia also contains an exceptionally large number of links embedded in the text. In order to make the situation more reminiscent of the Web in general, we divided the relevant material into cliques of 1-5 pages. Figure 1 illustrates the link structure within the material relevant to Task 1, the application of cryptography for improving the security of e-mail. The actual hyperlinks that were left in the documents are marked with solid arrows. For example, on the left side of the figure there is a clique of four pages containing information about the most widely used encryption software (*Pretty Good Privacy* and *GNU Privacy Guard*) and some closely related concepts (*Web of trust* and *Key signing party*). The dashed arrows indicate clear connections between the *contents* of the documents. The document titled *E-mail*, for example, discusses a wide variety of topics including the technical infrastructure, e-mail clients, and spam. There is also a section about privacy, which mentions PGP, GnuPG, and encryption along with a number of other distinctive terms. Identifying such *implicit links* is the main job of the proactive search engine interface.

Our work is organized in three stages. In the first stage we arranged a small-scale experiment to acquire the data needed for creating the testbed. We are currently engaged in the actual development of the proactive interface, guided by the kind of continuous testing described above. Finally, we will do a controlled experiment in order to evaluate the usefulness of the end result. The subjects will be divided into two groups, one of which will be provided with both keyword search and proactive search, and the other one with just keyword search. Using the same tasks as in the first stage, we will compare the performance of the groups by looking at the number relevant documents that the subjects are able to find with the available tools.

5 Related work

The idea of using a search engine to suggest related pages is fairly obvious and appears in numerous sources. Two implementations that serve as suitable examples for the present discussion are PowerScout [7] and Alexa [8].

PowerScout extracts keywords from the visible page, sends queries to a search engine, and shows an automatically updated list of results in a separate window. The queries can be focused and expanded by activating semiautomatically con-

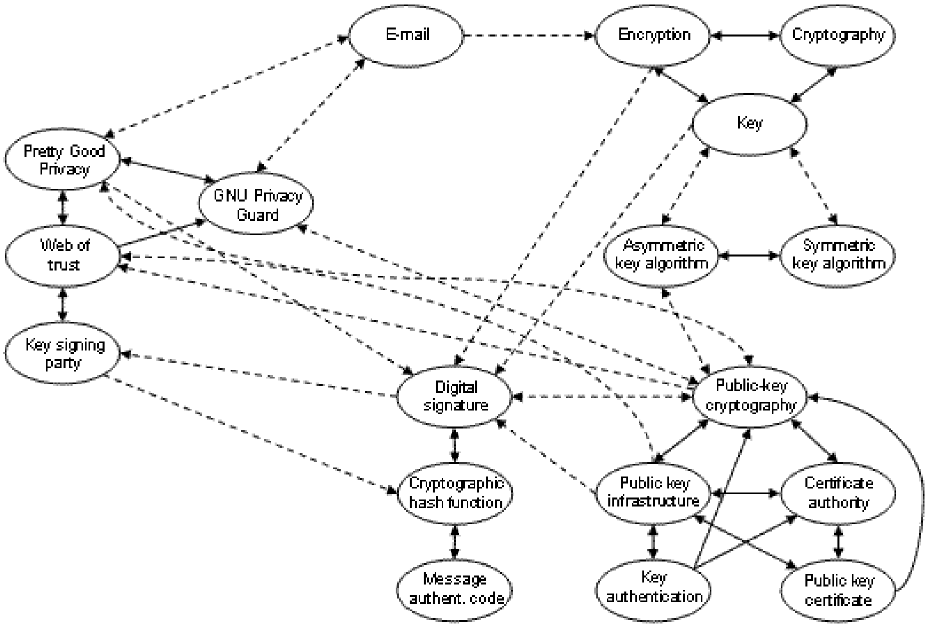

Fig. 1. The link structure within the set of documents relevant to Task 1. Some implicit links have been omitted for clarity.

structed *profiles*, which consist of supplementary keywords and represent the longer-term interests of the user. PowerScout provides an elaborate user interface, which enables e.g. manual editing of the profiles and the use of proactive queries as templates for explicit search.

Alexa is a large scale commercial system relying on *collaborative filtering.* It appears to the user as a toolbar that equips the Web browser with some additional features, including built-in keyword search, "Site Info" and "Related Links". Site Info contains usage statistics and reviews associated with the Web site that the user is currently visiting, and Related Links informs about other similar and potentially interesting pages. Both of these features are based on analysis of *navigation statistics* gathered from the entire user community.

Our approach has both similarities and differences compared to the previous work. Like PowerScout, our system tries to extract search engine queries automatically from the visible page. We are doing finer-grained analysis of the users' actions, however, as we are monitoring the *scrolling* of the pages in order to provide suggestions associated with specific *text fragments.* This may eliminate the need to construct explicit user profiles, resulting in a *reactive* system that provides effective support for *opportunistic navigation.* In case persistent user profiles turn out to be useful, our plan is to augment the queries with collaborative filtering techniques reminiscent of Alexa.

Relying on *content-based* search allows us to build the queries from longer fragments of text instead of individual keywords. The ranking of the results is based on the contents of the documents rather than their static link structure. These features of the underlying search engine hopefully enable it to handle automatically generated queries in a robust manner and provide results that better reflect the user's attention and interests.

In addition, we are in the unique position of being able to customize the search engine to the needs of proactive retrieval. Compared to the user interface, the search engine has superior information of the contents of the documents. Distributing the processing across several layers of the system rather than driving everything from the user interface opens up some interesting opportunities that to our knowledge have not been explored before.

6 Conclusions

We think of keyword search and proactive search as two complementary interfaces to the same underlying infrastructure. Although we are still at an early stage in our work with the proactive part, a substantial proportion of the system is up and running. It is scalable and robust, capable of dealing with millions of real-world Web pages.

Our first steps in the development of the proactive interface will rely on the testbed, however. Due to the (largely unknown) nature of the problem, it will be helpful, if not necessary, to have a clear understanding of the desired end result. This, along with the need for instantaneous testing, was our primary motivation for creating the testbed in the first place. Moreover, without a controlled environment it would be extremely tedious to tie the elements of proactive search engine together in a coherent manner.

The actual usefulness of the testbed depends on its validity as an environment for experimentation. Although we made deliberate attempts to limit the scope of the users' activities and simplify the measurement of the system's performance, we hopefully retained enough of the challenges involved in doing proactive information retrieval in more unstructured and open-ended settings.

7 Acknowledgements

This work was supported in part by the Academy of Finland under the Prima and Prose projects, and by the National Technology Agency under the project Search-Ina-Box. In addition, we received support from the IST Programme of the European Community under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

1. Tennenhouse, D.: Proactive computing. Communications of the ACM, 43(5), 43–75, 2000.
2. Rose, D.E. and Levinson, D.: Understanding user goals in web search. In Proceedings of the 13th international conference on World Wide Web, pages 13–19, ACM Press, 2004.
3. Teevan, J., Alvarado, C., Ackerman, M.S. and Karger, R.: The perfect search engine is not enough: a study of orienteering behavior in directed search. In Proceedings of the 2004 conference on human factors in computing systems, pages 415–422, ACM Press, 2004.
4. Ericsson, K.A. and Kintsch, W.: Long-term working memory. Psychological Review, 102, 211–245, 1995.
5. Liu, H., Lieberman, H. and Selker, T.: GOOSE: A goal-oriented search engine with commonsense. In Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, pages 253–263, Springer-Verlag, 2002.
6. Monsell, S.: Task switching Trends in Cognitive Sciences, 7(3), 134-140, 2003.
7. Lieberman, H., Fry, C. and Weitzman, L.: Exploring the Web with reconnaissance agents. Communications of the ACM, 44(8), 69–75, 2001.
8. http://www.alexa.com
9. Buntine W. and Jakulin A.: Applying Discrete PCA in Data Analysis. Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI'04), pages 59–66, AUAI Press, 2004.
10. Baeza-Yates R. and Ribeiro-Neto B.: Modern Information Retrieval. Addison Wesley, 1999.