Remembrance Agent: A continuously running automated information retrieval system

Bradley J. Rhodes
MIT Media Lab, E15-305
20 Ames St.
Cambridge, MA 02139 rhodes@media.mit.edu

Thad Starner
MIT Media Lab, E15-394
20 Ames St.
Cambridge, MA 02139 thad@media.mit.edu

Abstract

The Remembrance Agent (RA) is a program which augments human memory by displaying a list of documents which might be relevant to the user's current context. Unlike most information retrieval systems, the RA runs continuously without user intervention. Its unobtrusive interface allows a user to pursue or ignore the RA's suggestions as desired.

The Remembrance Agent

Most of a desktop computer's time is spent waiting: waiting for the user to hit the next keystroke, waiting for the user to read the next page, and waiting for the next packet to come down the network. The remembrance agent uses those wasted CPU cycles constructively by performing continuous searches for information that might be of use in its user's current situation. For example, while an engineer reads email about a project the remembrance agent reminds her of project schedules, status reports, and other resources related to the project in question. When she stops reading email and starts editing a file, the RA automatically changes it recommendations accordingly. These suggestions are presented in the form of one-line summaries at the bottom of the screen. Here they can be easily monitored, but won't distract from the primary work at hand. The full text of a suggestion can be brought up with a single keystroke.

Most applications for augmenting human memory, e.g. those developed by (Jones 1986) and (Lamming & Flynn 1994), have concentrated on recall-on-demand. For example, they can answer questions such as "when is that conference paper deadline?". However, they do not address situations where a user does not remember enough to realize he or she has forgotten something in the first place. This associative form of recall is what reminds a user that an important conference exists at all. The Remembrance Agent augments this continuous, associative form of recall.

Beyond simple memory enhancement, the RA also suggests alternative ways to organize knowledge. The connection between a suggested document and the user's current context may not be obvious to the user until pointed out. In this way the RA both helps the user organize their own datafiles into new categories, and helps create continuous brainstorming session where new ideas and possible connections are suggested.

Implementation

The RA front-end runs in elisp under Emacs-19, a UNIX based text editor which can also be used for applications such as email, netnews, and web access. The front end displays one-line suggestions along with a numeric rating indicating how relevant it thinks the document is. It also brings up the full text of suggested documents when requested. Every few seconds it collects the text within certain ranges around the current cursor position and sends this text to the information retrieval program.

The back-end is a program which, given a query-text, produces suggestions of similar documents from a pool of documents which are pre-indexed nightly. The current implementation uses the SMART information retrieval program, which decides document similarity based on the frequency of words common to the query and reference documents (Salton & Lesk 1971). While SMART is not as sophisticated as many more modern information-retrieval systems, it has the advantage that it requires no human preprocessing of the documents being indexed.

In the current version the RA utilizes two sources of suggestions. The first three lines print suggestions from the last year's worth of personal email (a little over 60 Megabytes worth). The last line prints suggestions from about 7 Megabytes of notes files entered over the past few years. The

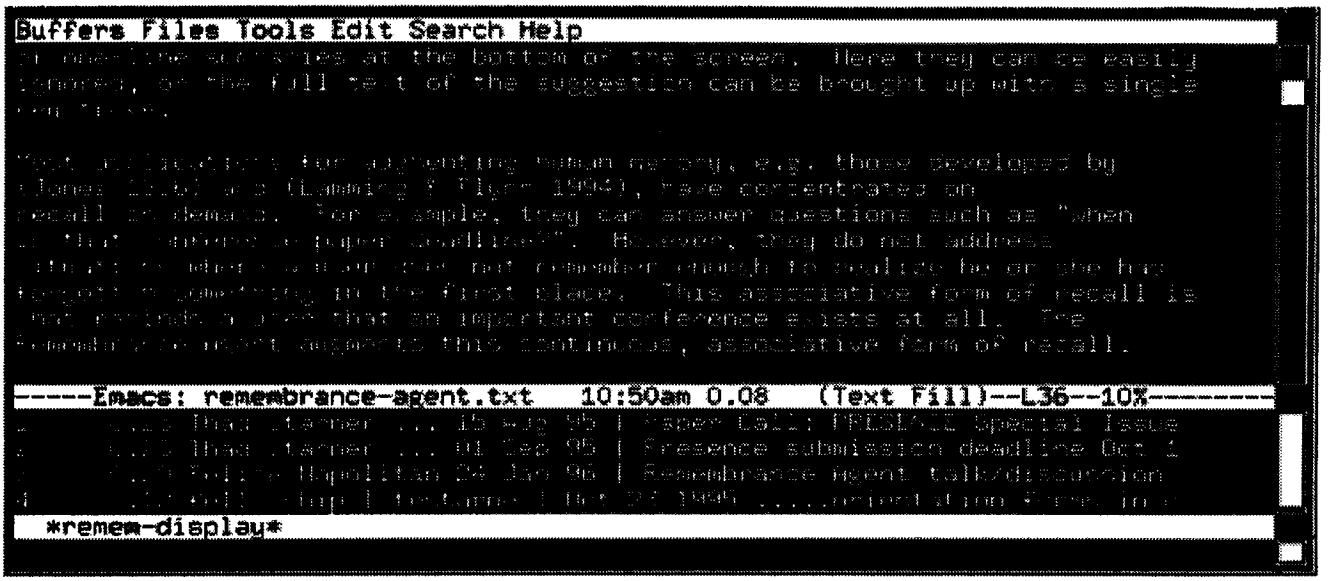

Figure 1: A screen-shot from the Remembrance Agent

short, one-line description of email messages contains who the message was from, when it was sent, and the subject line of the message. The one-line description of notes files contains the filename, the owner of the file, date last modified, and the first few words of the document.

The RA should be able to suggest documents similar to tangents in a document as well as those similar to the whole. At different times a user might be interested in suggestions relating to the person who's name was just typed, to the last paragraph typed, or relating to a much broader range. To achieve this, three "scopes" are defined for a document, centered around the current cursor location. One scope is dedicated to those suggestions relevant to the last thousand words (usually the entirety of the document). Another is dedicated to the last fifty words, and a third to the last ten words. Each scope accesses either the email or text database, and is allotted one or more display lines. The number of words each scope covers, the number of display lines each uses, the time between updates, and the database each points to are all customizable by the user.

Design Issues

There are three different ways of handling the timing of an automated task. The first is to perform a task only when specifically requested. Spell-checking a file and performing a web-search fall into this category. A second way is for an automated task to always lurk in the background, but to only act when a specific "trigger" occurs. Such programs include calendar programs that automatically tell you when you're going to be late for a meeting, and programs that alert you when you have new email waiting. Finally, there are tasks which are performed continuously, like a clock program or CPU-load meter. The RA falls into this third category.

Unlike programs which warn you about important meetings, it is usually not a major problem when the user misses suggestions presented by the RA. However, since the RA runs continuously, suggestions could quickly distract from the user's primary task if they attracted too much attention. For these reasons the RA's suggestions are kept unobtrusive. Suggestions are kept to a single line each, and are always printed at the bottom of the text-editor window. The full display area is also limited to a maximum of 9 lines, though it defaults to operating with only three or four. Finally, no color cues or highlights were used in the suggestion-display area.

It is often the case that just seeing the one-line description triggers the entire memory, with no need to bring up the full document. Also, in some cases the description line supplies desired information directly (for example, the last name of someone who sent email). For this reason, as much information as possible is packed into the short description line.

While suggestions are unobtrusive, it is easy to access the entirety of a suggested document. To view the document being suggested on a given line, the user simply types "control-c" and the display line-number. It is similarly easy to get back to the primary document.

As an additional feature, the user can explicitly ask for recommendations based on an input string. When this method is used to receive suggestions, all the display lines are used to present the top possible suggestions, and the scope

information is ignored. The scope information is similarly ignored when the user reads email. Email messages usually contains a single thought, so suggestions based on the entire email are usually more useful than those based on smaller ranges.

Using the system has shown that suggestions are much more useful when the document being suggested only contains one "nugget" of information, and when that nugget is clearly displayed on its one-line description. This "less is more" approach solves several problems. First, it allows the user to tell what a suggestion contains from its description without having to peruse the entire document. Second, a document with only one primary point is more likely to be a good hit. Documents which address several issues will rarely match the user's situation exactly, but will often partially match. Finally, if a suggested document *is* read, the shorter it is the quicker the user can get on with their primary work. Old email are a good length for RA suggestions.

Several of the applications described in the next section use the RA to suggest documents the user has never seen before. While this may be a powerful application, there are extra pitfalls when using someone else's database. The primary pitfall is that the user will not have the context associated with a given suggestion. This makes it far less likely that the user will recognize a document from its one-line summary, and it will be far more difficult for them to judge whether a suggestion is useful. Another problem is that the user won't necessarily know the original context in which a document was created. For example, they might not know that a set of instructions were made in jest, or were in a subsequent message shown to be incorrect. Similarly, documentation or lectures geared towards one kind of audience may be inappropriate for another, causing the original speaker's intention to be misinterpreted. Finally, there are privacy issues when using someone else's "memories." Even if you access someone's email files with their express permission, the people who sent that person mail might not have approved.

Future Applications

Reference advisor for technical papers

One application is to have the RA suggest technical reports on a given subject. When it recommends other researchers' papers, these could be referenced or tagged for later reading. When it recommends one's own old papers, these can be scanned for similar material which might be used in the new paper. Such a system could also recommend conferences where the call-for-papers is similar in content to one's own paper.

Knowledge transfer

One of the large difficulties facing industry is bringing new members of an existing work-group up to speed quickly. If a work-group created its own knowledge-base, new members could access the group Remembrance Agent. This would not only give the employee access to the group knowledge itself (as would any database), but also to the meta-knowledge of *when* particular information is relevant or valuable. Similar applications would exist wherever "just-in-time training" is required. While this application uses a knowledge base not familiar to the user at first, they will not suffer too greatly from the lack-of-context problem discussed above because the knowledge is focused on their current situation, namely their new position within the group. Any context they do not yet know, they need to learn anyway, and the RA will help them learn it.

Wearable computing

When running on a desktop computer, an RA can only guess the user's context based on the document they are reading or editing. However, the advent of wearable computing (Starner et al. 1995) will allow RAs to work with much more information and many more situations. Global Positioning Systems (GPS) will let the RA know where the user is, while camera and face recognition will let it know who they're talking to. With this extra information, a (greatly enhanced) RA could know that it is around lunch time, that according to camera input the user is with her lunch date, according to her appointment book she has an appointment with her boss in an hour, and according to her GPS she's downtown. From this information it could recommend several good restaurants known for their fast service that are within a few blocks of downtown, which would be agreeable to her lunch date as well.

Automatic Hypertexting

In the past year several on-line magazines have appeared, such as HotWired (HotWired), many of which have paper counterparts. One of the values added by the on-line versions of these magazines are hyper-linked text, where a reader can click on a word and get more information on that subject. A future RA could conceivably automatically turn normal email, netnews, or papers into hyper-linked documents, automatically linking hot-words to relevant background information.

Background checker

Another Web-based RA could perform background checks on people sending mail, referenced in papers, or recognized by a wearable-computer-mounted camera. Such an RA could automatically provide information on a person's

employer, job title, phone numbers, and profiles of their interests based on newsgroups they frequent. At the click of a button, the user could access that person's home-page for even more information.

References

HotWired. http://www.hotwired.com/

Jones, W. 1986. On the Applied Use of Human Memory Models: the Memory Extender Personal Filing System. *The International Journal of Man-Machine Studies* 25, 191-228

Lamming, M., and Flynn, M. 1994. "Forget-me-not:" Intimate Computing in Support of Human Memory. In Proceedings of FRIEND21, '94 International Symposium on Next Generation Human Interface, Meguro Gajoen, Japan.

Salton, G. ed. 1971. *The SMART Retrieval System -- Experiments in Automatic Document Processing.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

Starner, T., Mann, S., Rhodes, B., Healey, J., Russell, K., Levine, J., and Pentland, A. 1995. Wearable Computing and Augmented Reality, Technical Report, Media Lab Vision and Modeling Group RT-355, MIT

Wickens, C. 1992. *Engineering Psychology and Human Performance.* Scott Foresman Little Brown.