Just-in-time information retrieval agents

by B. J. Rhodes
P. Maes

A just-in-time information retrieval agent (JITIR agent) is software that proactively retrieves and presents information based on a person's local context in an easily accessible yet nonintrusive manner. This paper describes three implemented JITIR agents: the Remembrance Agent, Margin Notes, and Jimminy. Theory and design lessons learned from these implementations are presented, drawing from behavioral psychology, information retrieval, and interface design. They are followed by evaluations and experimental results. The key lesson is that users of JITIR agents are not merely more efficient at retrieving information, but actually retrieve and use more information than they would with traditional search engines.

In this paper, just-in-time information retrieval agents (JITIR agents) are introduced. JITIR agents (pronounced "jitter agents") are a class of software agents that proactively present information based on a person's context in an easily accessible and nonintrusive manner. They continuously watch a person's environment and present information that may be useful without requiring any action on the part of the user. The environment that an agent looks at is usually computational: e-mail, a Web page that a person is reading, or a document that he or she is writing. However, it can also be a person's physical environment as sensed by a hand-held or wearable computer. The information a JITIR agent provides can come from any number of preindexed databases of documents, e.g., e-mail archives, notes files, or documents from commercial databases such as the INSPEC collection of the Institute of Electrical and Electronics Engineers (IEEE) technical paper abstracts. A key feature of a JITIR agent is that it can provide information from an unordered set of documents; no hand-coding of documents or special domain-dependent techniques are required. This feature makes it easy to adapt systems to new domains and information sources.

The word "agent" has many different definitions. JITIR agents are "software agents" in that they are long-lived, watch an environment, and can take action based on that environment without direct user intervention. They should not be confused with "embodied conversational agents" or synthetic characters, which are graphical interactive characters that present a personality and interact with a user in an animistic way. They should also not be confused with distributed agent architectures or agent-oriented programming models, which are both architectures for software design rather than a class of applications. Unless specified, all references to the word "agents" in this document specifically mean JITIR agents.

The three necessary features of a JITIR agent are proactivity, the presentation of information in an accessible yet nonintrusive manner, and awareness of the user's local context. These three features make JITIR agents similar to search engines, alarms, and personalized help systems, although they differ from each.

Before the main discussion of the paper is begun, the similarities and differences are discussed below.

Search engines and structured knowledge bases such as Yahoo!** are inherently interactive: an information seeker has some query in mind and directly interacts with the program to obtain the desired information. JITIR agents, in contrast, are proactive. The user need not have a query in mind, or even know that information relevant to his or her situation exists. This proactivity has ramifications for the information retrieval techniques that can be used, because the "query" utilized is limited to what can be sensed in the environment. It also has ramifications for interface design, because proactively displayed information can be far more distracting than requested information.

When a cellular (cell) phone rings, it provides the information that a person is calling and may also indicate the identity of the caller by the tone of the ring. A ringing telephone is an alarm. It calls the user away from whatever task is currently being performed and cannot easily be ignored. If the cell phone is turned off, the caller is often forwarded to voice mail. A silent cell phone is extremely nonintrusive, but without the ringer it is necessary to call the voice-mail service directly to determine whether anyone has called. The information about who called (if anyone) is less accessible than when it was indicated by the ring. JITIR agents are designed to operate in between these two extremes by presenting information in such a way that it can be ignored, but is still easy to access should it be desirable. Rather than presuppose whether or not a particular piece of information is important or urgent, JITIR agents allow the user to decide whether to view or ignore it, depending on the user's current task and level of cognitive load.

Notification systems such as newspaper clipping services and alerts are proactive, but the information they present is based on events outside of the user's local context. For example, an alert might be triggered whenever a new piece of e-mail arrives, a stock price goes below a certain threshold, or news that fits a user's personal profile hits the news wire. The notifications are designed to pull a person out of his or her current context (task) and provide information about a different context that might require attention. The urgency of a notification can range from the immediacy of a fire alarm to a news briefing that is announced but intended to be read whenever convenient.

Notification systems present information from a rapidly changing source (e.g., current stock prices), based on relevance to a mostly static user profile. JITIR agents are the reverse: they provide information from a mostly static source (e.g., e-mail archives) based on relevance to a user's rapidly changing local context. Information provided by an agent is not meant to pull a person out of his or her current context, but rather to add additional information that might be useful *within* that context.

In summary, JITIR agents are similar to search engines, alarms, and notification systems, but none of these systems has all three features necessary for JITIR agents: proactivity, a nonintrusive yet accessible interface, and attention to local context.

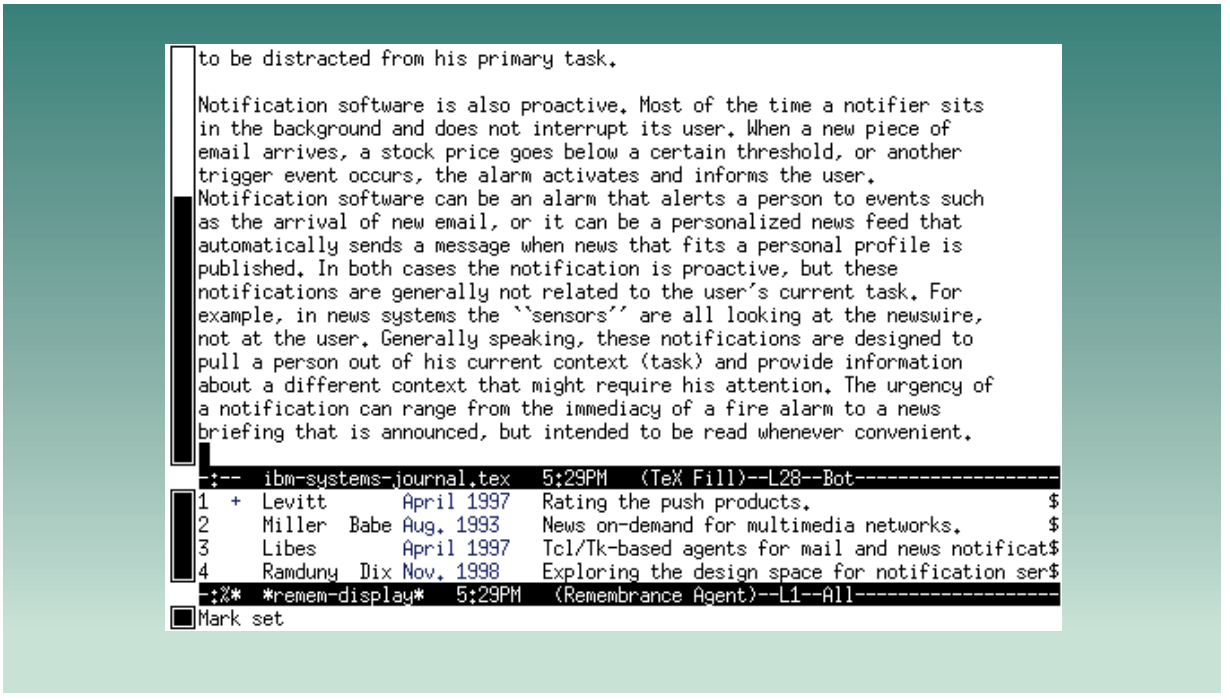
Note that automatic help systems, such as the Microsoft Office Assistant**, fit the definition of a JITIR agent. However, these systems are domain-specific; they only provide information from a specialized help database, using information-retrieval techniques that are specific for that particular help domain. Although no system can support every possible domain and environment, the JITIR agents described in this paper are designed to provide information from a wide variety of sources using techniques that can be applied to a wide variety of task domains.

Implementations

Three JITIR agents have been deployed in the course of this research. The first and oldest is the Remembrance Agent (RA), an agent that operates within the Emacs text editor, a program developed at the Massachusetts Institute of Technology (MIT). The second implementation is Margin Notes, a Web-based agent that automatically annotates Web pages as they are being loaded into a browser. The third system is Jimminy (also called the Wearable RA). Jimminy presents information via a head-mounted display attached to a wearable computer, based on a person's physical environment: where the person is, who he or she is talking to, the time of day, etc. The system demonstrates the multidimensional aspects of the information retrieval system used and how these agents can be applied "off the desktop."

All three systems use the same information retrieval back end, called Savant, which was developed at the MIT Media Laboratory especially for this research. Savant uses a template structure to identify file types and parse out individual documents and fields. For example, it can identify a mail archive file and index

Figure 1 RA main display



individual e-mail documents by the from field, subject line, date, and body of the message. Savant can also handle multiple data types including the GPS (Global Positioning System) and date as well as raw text, and can be easily extended to include more types of fields and similarity metrics.

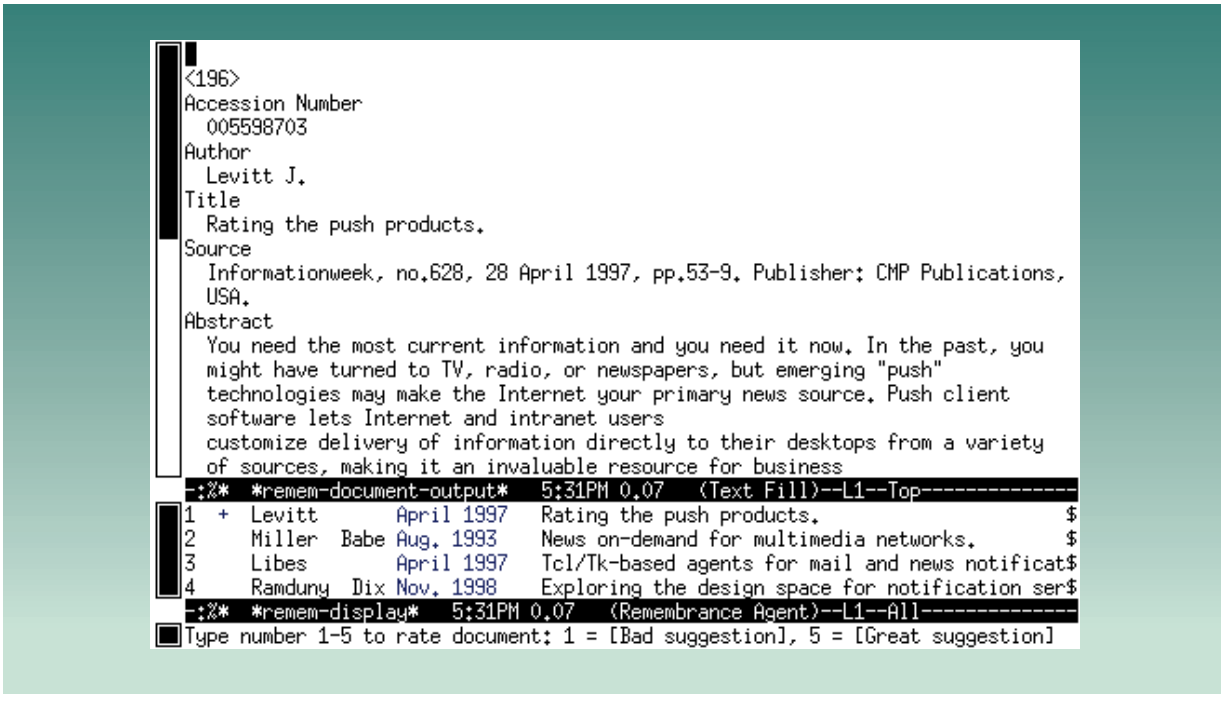
The Remembrance Agent. Emacs is a popular text editor for the UNIX** operating system. Although it is often used for traditional text-editing tasks such as writing papers or computer code, the editor is powerful enough that it is also used for reading and writing e-mail and network news, and even browsing the Web. Emacs supports the use of multiple buffers, with each buffer containing a different file.

The RA[1] continually presents a list of documents that are related to the current document being written or read. These suggestions appear in order of relevance within a special display buffer at the bottom of the Emacs window. When the user is typing, reading e-mails, or otherwise changing his or her environment, the list is updated every few seconds. Suggestions from multiple databases can be listed in the display buffer, each with a certain number of lines. For example, the system can be configured to display suggestions from e-mail archives in the first four lines and suggestions from note files in the next two lines. The display can also show different "scopes" from the same database, e.g., the first few lines can show suggestions based on the past 20 words, whereas the others can show suggestions based on the past 500 words.

Figure 1 shows a screen shot of the RA when writing an earlier version of the introduction to this paper. In this case, the documents being suggested come from a subset of the INSPEC database of paper abstracts (about 150000 citations). The suggestions are all for papers that might be relevant to the section being written. The full Emacs window is larger in normal operation, which means a larger ratio of editor lines to RA-suggestion lines.

The summary lines are a combination of fields in the document and are designed to give the user an indication of the content of the document as quickly as possible. These summary lines can be customized for individual databases. For example, suggestions from the INSPEC database display author, date of

Figure 2 Emacs RA result screen



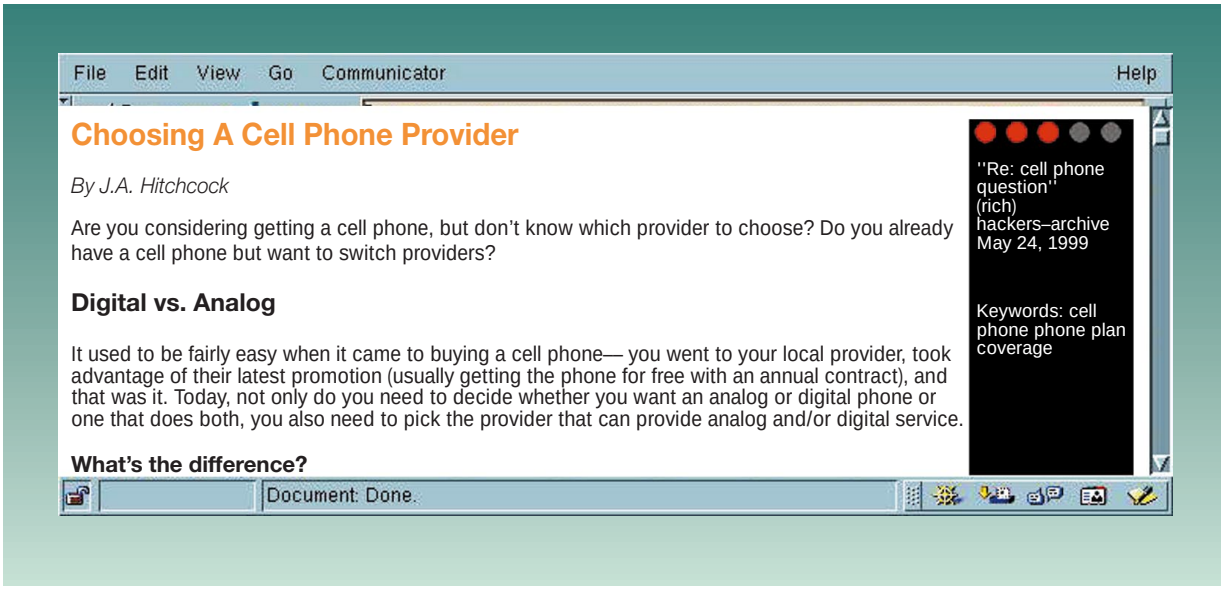
publication, and paper title. Articles from *The Boston Globe*, in contrast, use a longer field to show headlines and show publication date, but do not show the author of the article. All summary lines also include a relevance score, consisting of zero, one, or two plus signs. By default, if a suggestion is below a minimum threshold, it is not displayed, and "No suggestion" is shown instead. It is also possible to configure the system to display below-threshold suggestions with a minus sign as the relevance.

Right-clicking on a suggestion causes a small pop-up window to display the keywords that lead to the abstract being suggested. These keywords are also displayed to the far right of the suggestion line, although due to space constraints they are only visible when the user has a wide display window. To see the full text being suggested, the user types a keyboard shortcut or clicks on the desired line number. The full text then replaces the current display buffer, as shown in Figure 2. By default the RA will not make suggestions based on a suggestion that has been retrieved, though this is customizable.

The system can also be used as a normal search engine. Left-clicking on a field (e.g., the author field) automatically runs a search for other documents associated with that person or field. A full search form is also available to perform manual queries. These features allow two-way dialog between user and RA. For example, in the example in Figure 1, one may not care about the "News on-demand . . ." paper but might want to know about other papers on news. Left-clicking on the subject field brings up summaries for other documents with the same words in the title. These suggestions may then lead the user to perform a manual search, which in turn may lead to more suggestions.

Different suggestion databases can be associated with specific buffers or buffer types. For example, the system can be configured to automatically draw suggestions from an e-mail archive when reading or writing e-mail, and from the INSPEC database whenever writing a paper in the LaTeX formatting mode. These defaults are set by hand in a configuration file. Databases can also be switched manually with a keyboard shortcut. Database changes are sticky: If the database is switched once for a particular buffer, then revisiting that buffer will automatically switch to that database thereafter.

Figure 3 Margin Notes screen shot



Margin Notes. Margin Notes[2] is a JITIR agent that automatically rewrites Web pages as they are loaded, adding hyperlinks to personal files. As a Web page is loaded, Margin Notes adds a black margin strip to the right of the document. Like the RA, it then compares each section of the document to preindexed e-mail archives, notes files, and other text files, based on keyword co-occurrence. If one of these indexed files is found to be relevant to the current section of the Web page, a small "suggestion note" is included in the margin next to the relevant section. The note contains a quick description of the suggested text, a series of circles representing the relevance of the suggestion, and a link to obtain more information. The suggestion note consists of a subject, date, and author for the suggested text, though the exact makeup of the note is customizable based on the database.

Figure 3 shows a screen shot of a Web page on cellular phone service plans, annotated by Margin Notes. This example is a page and annotation that came up during normal use of Margin Notes, though the relevance of the annotation is better than the typical annotation. The database used for this example is the collection of Media Lab e-mail archives since 1988, a total of over 180000 messages. The suggested document, listed in the black margin to the right, is e-mail sent to the "hackers" mailing list and gives personal experiences with cellular service in the MIT area. The circles at the top of a suggestion are filled in with red to indicate the relevance of the suggestion.

Placing the mouse over a suggestion note produces a list of the five keywords that were most important in deciding the relevance of a suggestion to the annotated Web page section (these are also shown in the figure). Clicking on a suggestion note creates a new browser window that displays the full text of the e-mail, note file, or text being suggested. The suggested page also has feedback buttons used for evaluating the system.

Because Web pages will often cover many different subjects, Margin Notes segments the Web pages it annotates based on HTML (HyperText Markup Language) tags. Each section receives its own annotation, assuming the annotation is over a threshold relevance score. The exception is the first annotation on each Web page, which is based on the entire page instead of a single section. This gives both a general annotation as well as a specific, focused view. Margin Notes uses the location of the annotation to indicate scope: annotations appear at the top of the section to which they are relevant. Thus it is analogous to marginal notes in traditional print. The black margin strip achieves "branding"; all text to

the left of the margin is the page being annotated, all text within the margin is placed there by Margin Notes.

Jimminy. Both the RA and Margin Notes provide potentially useful information based on a person's computational environment: his or her e-mail, documents, Web pages being viewed, etc. Jimminy provides information based on a person's physical environment: his or her location, people in the room, time of day, and subject of conversation.[3] Processing is performed on a shoulder-mounted "wearable computer," and suggestions are presented on a head-mounted display.

The ultimate goal is that all information about the wearer's physical environment will be available to Jimminy through automatic sensors. However, the focus of this research is not sensor technology but rather what can be done with that technology once it is available. To this end, Jimminy is a general architecture that can use plug-ins for any sensor that can be attached to a wearable computer. Information that is not provided by sensors can be entered into the system by hand. The currently implemented system has been demonstrated with passive sensors that detect a person's physical location, people in the room, and the time of day. The general topic of a conversation is entered by hand in the form of real-time notes.

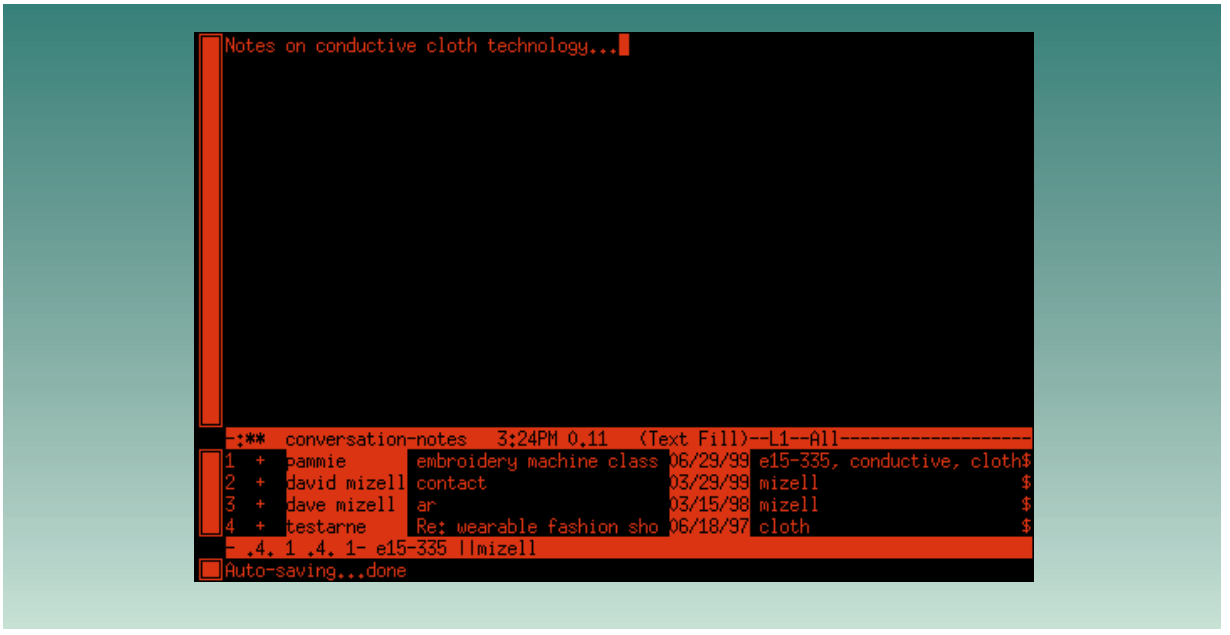
The hardware for the system is the "Lizzy" wearable computer designed by Thad Starner,[4] which consists of a 100-MHz 486 processor running the Linux** operating system, a head-mounted display, and one-handed chording keyboard. The entire system fits in a small shoulder bag. The display is the "Private Eye**" made by Reflection Technology. Display resolution is 720 $\times$ 280 pixels, monochrome red with one level of dimness. This display gives a crisp 80-column by 25-row screen with a virtual image seeming to float in front of the wearer. The keyboard is the "Twiddler**," a one-handed keyboard made by Handykey Corporation that uses a 17-button system, on which multiple keys can be struck at once to access all the symbols possible on a normal keyboard, plus extra combinations for macros. Average typing speed is about 35 words per minute using the Twiddler, although Starner has been clocked at up to 60 words per minute.

The wearable computer also includes several ways to sense the outside world. When outdoors, a GPS is used to detect the wearer's location. When indoors, a 418-MHz AM radio receiver detects unique radio beacons that have been placed around the Media Lab.[5] An alternate system uses IR (infrared light) instead of radio, which gives a finer control over where a beacon will be detected.[6] Beacon numbers are converted to room numbers via a static lookup. By putting these beacons into name badges, the wearable can also detect who is currently in the same room as the wearable user. This method is essentially identical to the active badge system designed at Olivetti.[7] However, because people do not generally wear these active badges, the people sensor is only used for demonstration purposes. The wearable computer also has a 1.2-Mbit wireless network connection that is used for communications and receiving information from sensors not directly attached to the computer. Sensors and Jimminy are interconnected using Hive, a distributed agent architecture developed by Nelson Minar.[8]

With recent improvements in sensor technology and in the processor speed of wearable computers, it is expected that other types of sensing technology will soon become available. One such technique is ASR (automatic speech recognition), which is now accurate enough so that information retrieval on a database of raw audio news stories can be performed with over 55 percent precision.[9] This percentage is close to the level of performance that would be needed to automatically generate queries for Jimminy by transcribing a person's natural conversational speech. Another technique is vision-based automatic face recognition,[10] which could be used instead of active badges to let the wearable know what other people are in the room.

The wearable computer serves two main functions. First, it is used as a general note-taking system. In conversations and lectures, notes are usually touch-typed using the one-handed keyboard while maintaining eye contact with the person speaking. The head-mounted display is occasionally viewed to see what has just been typed. Any note written using the wearable computer can be tagged with people present, subject, location, and time stamp using a single key combination. Over 850 notes have been taken and annotated on the wearable computer by this means over the course of four years. They range from notes on classes and conversations at conferences to notes on dance steps. The second major use of the wearable computer is to retrieve notes and information anytime, anywhere. Reading and understanding information on the head-mounted display is a more attention-demanding task than note-tak-

Figure 4 Jimminy screen shot



ing. It is also more obvious to observers that the user is distracted: the user's eyes are clearly focused on the screen, and it is difficult to speak and read at the same time. For these reasons information tends to be looked up during pauses in conversation or in lecture situations.

The interface for Jimminy is based on the RA but differs in a few important ways. First, screen real estate is scarce for a wearable computer, so suggestions are presented in abbreviated form. Second, the features of the environment that are currently being sensed are listed on the mode line. Having this information is important because sensor data may be incorrect, and the user also needs a reminder about what environmental information was last entered by hand so it can be updated. Third, keys are defined on the chording keyboard to increment or decrement the bias on different field types. For example, one could set biases so that the person field is twice as important as other fields when finding useful documents. The biases are listed on the mode line as well. Finally, the system will modify the bias for certain features based on recent-modification times. For example, if the wearer of the system enters a new room, the bias for room location is temporarily increased by three. After a minute in the same room, the bias is returned to base-line levels.

Figure 4 shows a screen shot of Jimminy as it would appear on the head-mounted display. The top 80 percent of the screen is reserved for notes being entered or read, plus the standard Emacs mode line giving time and information about the file being edited. The next four lines show suggestions based on the wearer's current context. The display is the same as the RA except formatted for the 80-column monochrome display. The bottom mode line shows a condensed view of current biases for location, subject, person, and current text being typed, followed by the context currently being used. The actual head-mounted display is bright red on black with 720 $\times$ 240 pixel resolution, with the number of characters and the aspect ratio shown in the figure.

For the example scenario of Figure 4, imagine the wearer is talking with David Mizell, a fellow "wearables" researcher from Boeing, and has just entered Media Laboratory room E15-335, which is where the automatic embroidery machine used in one of the research projects is kept. This example is hypothetical, although the screen shot is of the actual system using the real database of notes written on the wearable. The mode line shows the current location and people in the wearer's environment and shows that the bias for location and people present is four, and the bias for subject and current text being typed are

both one. The periods around the location and person biases indicate that they have temporarily been raised because these features changed recently. In one minute it will go back to the hand-set value of one. Biases can also be set using the chord keyboard. The first suggestion "embroidery machine class" is a note that was taken during the training class for the embroidery machine. As can be seen in the keywords section (far right), the suggestion is based mainly on the room number, but also on some of the words in the notes being typed. The second and third notes are about David Mizell, the first being his entry in the wearer's contact Rolodex** file and the other being notes about his talk on augmented reality at a conference in 1998. The final suggestion is an e-mail note about the wearable fashion show for which the conductive cloth technology was being developed at the Media Lab, based on keywords that have been typed into the notes area.

Savant. All three implemented JITIR agents use the same back-end system, called Savant. The front end senses the environment (that is, the document or e-mail being written, the Web page being viewed, or the physical environment of the user of a wearable computer) and sends that information in text form to Savant as a "query." Savant then works as an information retrieval engine: given a query, it produces a rank-ordered list of preindexed documents that best match the query. Savant consists of two programs: *ra-retrieve* performs information retrieval based on a query, and *ra-index* creates index files so retrieval can be performed quickly. Indexes can be created from generally useful sources such as a collection of newspaper or journal articles, organization-wide collections such as office memos, or from personal sources such as e-mail and notes. Previous versions also allowed pages to be indexed directly from the Web. Documents are usually reindexed nightly to incorporate new changes and additions.

The power of Savant comes from a strong template-matching system that can recognize documents, parse out fields, and index them all automatically. For example, if pointed at a top-level directory of heterogeneous files, it will recognize and parse e-mail archives, HTML files, LaTeX files, notes taken on the wearable computer, paper abstracts from the INSPEC database, and raw text, while ignoring other file formats. It will also break e-mail archives into individual messages. This means indexing can be performed completely automatically with no hand annotation. Different fields from documents are identified and indexed separately. For example, the from fields of e-mail archives are indexed as *people*, whereas the title fields of HTML documents and the subject fields of e-mail messages are indexed as *subjects*. Different field types can have different similarity metrics. For example, a collaborator at British Telecom has developed a version of the RA that can find documents that are related to a current location based on GPS coordinates.[11] If a document is similar to a person's current context based on more than one field (e.g., if both the "from" field and body of a message partially match), a linear combination of the similarities are used based on weights defined in the templates. The template system is also used for queries, so fields can be parsed out of e-mail being written or Web pages being read. Templates are hard-coded into Savant, but are designed to be easily modified or added with a recompilation of the source code.

Automatically generated queries tend to contain extraneous text, such as signature lines and e-mail headers, that is not useful for retrieval. Indexed documents will likewise have HTML markup and headers that will dilute the value of important data when selecting documents. To address this problem, each template can associate a filter bank with each field. A filter bank is an ordered list of Perl-style regular expressions that match text that should be removed from the field before parsing. For example, filters associated with the e-mail *body* field recognize and remove e-mail signature lines, headers from included files, and common lines such as "Begin forwarded message." Filters associated with the e-mail *person* field remove all information except for the user name, and filters associated with all fields in HTML documents remove hypertext tags and comments.

Although Savant can simultaneously use different similarity metrics for different fields, the most common metric is for similarity between two texts. The particular algorithm used is the Okapi version of the Term Frequency/Inverse Document Frequency (TF/iDF) algorithm, a fairly standard benchmark text retrieval algorithm.[12]

Theoretical issues

There are three primary questions being addressed by this research, ranging from the fields of psychology and decision sciences to document retrieval to interface design and cognitive science. An overview of the issues surrounding these questions is presented below. For a full discussion, interested readers are referred to the dissertation by Rhodes.[13]

How does the use of a JITIR agent affect the way in which a person uses information? Imagine an extremely paranoid executive who wants to be sure he reads every piece of information related to his work. To make sure he does not miss anything, he searches his old e-mails and office memos after every paragraph he writes. This would imitate the effect of a JITIR agent. In fact, it would be better than a JITIR agent because he could perform his searches more precisely than could any automated process.

Of course, no one is as meticulous as the executive described. Sometimes a person wants a particular piece of information, and in this case a search engine is the appropriate tool. Other times a person will not bother to perform a search because of a lack of time, because the information he or she already has is "good enough," or because the person expects a search will not turn up anything useful.

Such action (or inaction) is in keeping with *Zipf's Principle of Least Effort*,[14] which states that people will try to minimize their total future work, *given their best estimates at the time*. Payne, Bettman, and Johnson expand on this principle, arguing that people choose decision-making strategies based on multiple goals, including goals of accuracy and the conservation of limited cognitive responses.[15] This framework is based on anticipated accuracy versus anticipated effort: a decision maker assesses the benefits and costs of available strategies and chooses the strategy with the best cost/benefit trade-off. The effort-accuracy framework has been used to explain why people behave differently using different kinds of information displays in a computational environment.[16] It can also be used to explain why people make different purchases depending on whether per-unit prices in a grocery store are listed on a single sign or only listed under each product.[17]

Studies in computer response time indicate that small increases in the effort required to perform a task can have large effects on whether a person will bother acting at all. For example, Robert Miller argues that for many tasks, more than two seconds of response delay is unacceptable and will result in fewer uses of a particular tool, even at the cost of decreased accuracy.[18] He also argues that there is not a linear decrease in efficiency as response delay increases; there are response delay thresholds beyond which sudden drops in mental efficiency will occur. Miller is primarily discussing system response *delays*, but the same short-term memory limitations he discusses also apply to performing subtasks that distract from a primary task. For example, when performing a search for information about a digression, a person needs to use short-term memory to keep his or her place in the larger framework of the task. The amount of short-term memory required, and thus the amount of effort required in the task, will depend on a number of factors including the complexity of the digression, the amount of time required to complete the task, and the similarity of the digression to the primary task.

When applied to the information search domain, these theories suggest that if the cost of finding and using information is more than the expected value of that information, then the search will not be performed. This result could be the case for several reasons. First, the desired information may not be important enough. For example, the searcher might think he or she "remembers it well enough," or there might be little reward for accuracy. The person might also think a search will not be fruitful, and thus the expected value is low even if an unexpectedly good result would be worthwhile. Finally, the person could be under time pressure or dealing with a poor search interface, and thus the effort required for a search is too costly.

JITR agents greatly reduce the cost of searching for information by doing most of the work automatically.

JITIR agents greatly reduce the cost of searching for information by doing most of the work automatically. The downside is that queries are automatically generated and therefore will not be as exact as would a human-generated query. This low-cost search is more than just a time-saver, it has the qualitative effect of providing information in situations where an individual is not personally willing to perform a search. In terms of an effort-accuracy trade-off, a JITIR agent is a resource for retrieving information that would otherwise be too costly to search for or where a search by other means has a low expected benefit.

How can a JITIR agent automatically find information that would be useful to a person by looking at that person's current environment? The perfect retrieval engine for a JITIR agent would instantly be able to know whether a piece of information would be useful to a person. The next best thing would be an engine that could know a person's task, what information he or she already knows, the thoughts he or she is currently thinking, and how he or she processes new information. With such information, an agent could easily deduce whether information would be useful. Unfortunately, we have neither the ability to prognosticate nor read minds. An agent must "make do" with whatever limited information it can sense automatically from a person's computational or physical environment.

Because it is impossible to directly sense whether a document will be useful or not, features of the environment that are detectable must be used as predictors, or proxies for usefulness. For the JITIR agents presented here, the *similarity* of a document to the person's current local context is used as the proxy for usefulness. The more similar it is, the more useful it is likely to be.

Although Savant is designed to use many similarity metrics, the implemented systems all rely heavily on text-retrieval techniques. The retrieval problems for JITIR agents differ from traditional text retrieval as described in the following subsections.

Lack of priors. A person chooses information resources based on certain needs at the time. For example, if a doctor needs a medical reference, he or she will use the National Institutes of Health search site. If the doctor wants to know where to play golf, he or she will browse the tourism board database. The choice of information source is one of the best indications of a person's current needs. In terms of probability, the fact that a person chose a particular database gives a strong prior (bias in probability) that information in that database will be useful. Usually JITIR agents do not have such a direct indication of what information might be valuable. Although one could access a combined database of all information that might be valuable, this solution neither scales nor retrieves quality documents.

Of course, some priors exist even with JITIR agents. The kinds of information that can be provided by an agent will be implicitly constrained by its interface, the sensors it uses, and the people who use it. For example, an agent embedded in a word processor can be expected to provide information that might be related to anything typed in that application, but not in other applications. This constraint provides some natural limits on the kinds of tasks being performed, though usually not as many as are enjoyed by a specialized search engine. It is also sometimes possible to use features of the task environment to make a good first guess at the kind of information that might be most useful. For example, the RA uses the Emacs editing mode to tell whether a person is reading e-mail, writing code, or writing a paper. This information is used to choose between archived e-mail, a code library, or paper abstracts from the INSPEC database.

Multiple topics. In a manual information system a person usually searches for one piece of information at a time. A person's environment will usually relate to several different subjects. For example, the discussion in this paper ranges from information retrieval to interface design to specific implementations of JITIR agents. E-mail will often cover even more disparate topics within the same message.

Sometimes multiple-subject queries can be an advantage. Any documents that match more than one of the subjects (e.g., in this case relating to both information retrieval and interface design) are likely to be useful documents, and documents relating to just one subject represented in a query might still be useful. However, multiple-subject queries can also cause traditional IR techniques to miss documents that are very relevant to one particular subject in favor of documents only somewhat relevant to all subjects represented. Furthermore, parts of the query might not be useful to a retrieval engine at all. For example, a signature line at the bottom of an e-mail may not contain any information relevant to a person's current task or interests. This information must therefore be removed or otherwise ignored.

Different criteria for evaluation. In the text-retrieval field, algorithms are typically evaluated based on whether the documents returned are relevant to the given query. It is assumed that the query is a good indication of the user's interests. JITIR agent queries are automatically created, so relevance is not good enough. To evaluate the information retrieval algorithm of a JITIR agent, one needs to show that the hits returned are useful to a person given a current task. Although relevance may correlate with usefulness, the two are not the same. For example, a citation from the INSPEC database could be relevant to a paper a researcher is writing but still be useless

if the suggested document is already known. Although relevance can be evaluated given only a query, usefulness must be evaluated for a given person in a given task environment. This requirement makes it more difficult to generalize results.

Queries are longer. In the past five years the IR field has been attempting to produce relevant documents based on shorter queries. This trend has been spurred by the needs of Web search engines, where the average query length is less than three words.[19] Many of the techniques developed have been ways to perform query expansion, where a short query is automatically augmented with words appearing in the best ranked documents of an initial probe search.[20] With JITIR agents, the environment provides a large amount of data that can potentially be a part of a query, so query expansion is less important.

Both indexed documents and queries are multivariate. Both documents being suggested and queries themselves will often contain people's names, dates, subjects, abstracts, locations, phone numbers, and a host of other information types. Although manual queries can also contain such information, they are often sparser because of the difficulties of entering large amounts of data.

JITIR agents need both ranked-best evaluation and filtering. Search engines normally produce a ranked-best list of hits for a given query. The absolute relevance score of a hit is not important as long as this relative ranking is correct. JITIR agents require a combination of ranked-best evaluation and filtering. They need to display the top few hits, but also need to display an absolute confidence in the relevance of a suggestion and to potentially suppress the display of low-quality suggestions. This need is similar to the problem faced by text-filtering systems such as automatic newspaper-clipping services. However, these systems assume that a stream of documents is compared to a long-lasting set of query profiles, such as a list of keywords expressing user interests.[21] The queries in JITIR agents are not long-lasting, so most of the machine-learning techniques used by these text-filtering systems cannot be applied.

High precision is required. Information retrieval researchers often talk about the trade-off between precision (making sure all suggested documents are of high relevance) and recall (making sure all relevant documents are suggested). Because JITIR agents largely play a supporting rather than a primary task role, they usually should not suggest more than a few documents. More suggestions, even if all of them were relevant, would be too much of a distraction. For this reason, the information retrieval algorithms for JITIR agents should tend to favor precision over recall.

How should a JITIR agent present potentially useful information? The most important design constraint for JITIR agents is that reading provided information should be a secondary task. Unlike users of a search engine, JITIR agent users are not actively seeking information being suggested. They are less tolerant of distraction from their primary task and are less willing to dig through suggestions to find useful information. Furthermore, even if a suggested text is highly relevant to a user's current context, the person may not be interested in it. The text could already be known, the user may not wish to be distracted, or may simply not want any new information. For this reason an interface must have a low cost for false positives. It must be an *ignorable* interface, although not so ignorable that it is never used.

The design of an ignorable interface must take into account two limitations on cognitive processes: *focused attention* and *divided attention*. Focused attention is the ability to attend to intended stimuli and tasks while ignoring others. The other side of the coin is *divided attention*: the ability to switch between tasks or stimuli. Generally speaking, it is easier to focus one's attention when the features of the environment being attended are dissimilar to distractions.[22] For example, it is easy to listen to talk radio while driving because driving is largely visual and spatial, whereas listening to the radio is auditory and verbal.

Unfortunately, there is a trade-off between focused and divided attention: the same similarity in display that makes it harder to focus on one stimulus and ignore the other makes it easier to switch focus between two stimuli. This trade-off leads to the *proximity compatibility principle*, which states that high display proximity (similarity) helps in tasks with similar mental proximity, and where information is related and needs to be treated together.[23] For example, military pilots use head-up displays to place annotations visually on top of enemy and friendly aircraft. This use of spatial proximity helps link the annotation to the object but can make it more difficult to process only one without the other.

Divided attention is also easier when switching between tasks because it does not require a large shift in the contents of short-term memory. This situation is obviously the case when one of the tasks has low memory requirements. For example, turning on a light while talking on the phone requires little shifting of memory because the location of the light switch can be seen: it is knowledge in the world. However, finding an object or reading e-mail while on the phone requires more swapping of short-term memory between tasks, and the phone conversation will probably suffer. Short-term memory also requires less swapping if the two tasks share similar mental models, or schema. For example, several schema will be shared by tasks that relate to the same general task or subject.[24]

It is important to be able to distinguish a suggestion from the environment.

JITIR agents need to allow persons to focus their attention on their primary task, and also to divide their attention between the primary task and the information provided by the agent.

To make focused attention easier, a JITIR agent should use different modalities or channels than are used by a person's primary task. Such use is especially important when the primary task is demanding. For example, Jimminy is designed to give information to persons as they engage in conversation or listen to lectures. In these environments the auditory modality is primary, so Jimminy uses a visual display for output.

It is also important that suggested information is not linked to parts of the environment to which it is not relevant. For example, if an agent is giving information related to a text editor, the display should not be near the Web browser, nor should it have a color scheme or layout that associates it with the Web browser. Doing so would encourage checking the JITIR agent output at times when the suggestions are almost guaranteed not to be related to the current task.

On a related note, it is important to be able to distinguish a suggestion from the environment. For example, it must be clear that a suggestion on a Web page comes from Margin Notes and is not actually a part of the original Web page being viewed. One method is to ensure that suggestions appear out-of-band, for example, in a separate window or different modality. Another method is to ensure that suggestions look nothing like the user's other context. For example, annotations in an augmented reality system are never mistaken for the real world because the resolution of computer graphics is not yet high enough. The third method is branding: ensuring that annotations have a unique "look and feel" that sets them apart. For example, annotations could use a different font, color, location, or modality than the information being annotated.

To make divided attention (task switching) easier, a JITIR agent should display information in a way that is congruous with the parts of the environment to which it relates. For example, it is likely easier to look at suggestions from an e-mail archive when reading or writing e-mail because the two formats have the same mental model. Similar mappings between suggestion and the context to which it is relevant can be achieved with color and font. Probably the most effective way to link information is to use spatial proximity: put information near what it is about. In the Remembrance Agent, the suggestion display is in a buffer within the editor window rather than a separate window. This placement links the RA with the particular text being edited and keeps it from being linked with other applications running on the desktop. In Margin Notes, annotations appear to the right of the paragraph or section to which they relate. Moving the scroll bar in the Web browser moves the annotations as well, increasing the mental linkage between the two.

The Margin Notes example reveals another use of spatial proximity: It can indicate to which part of a user's context a suggestion is relevant. Even if a person's full context is constrained to a single Web page, it should still be apparent whether a suggestion is relevant to a single paragraph, a section, or the entire Web page. Spatial proximity is a good way to indicate this information, although when this is not possible, the indication can still be by fiat, e.g., by declaring that suggestions are chosen based on relevance to the whole body, or by indicating the scope of relevance in the suggestion in some other way.

Even when the interface is designed to help divided attention, a JITIR agent will still increase cognitive load. When the information provided is valuable, the trade-off is worthwhile, but false positives will still be a problem. False positives are also guaranteed: a JITIR agent will never reach 100 percent useful information without perfect information about a person's mental state. Given this limitation, information should be displayed to minimize the cognitive and perceptual load the user must undertake to evaluate the quality of information provided.

One display technique is what we call a "ramping interface," where information is conveyed in stages. Each stage of a ramping interface provides a little more information, at the cost of a little more attention required to read and understand it. The idea is to present useful information, while at the same time allowing a person to detect bad information and bail out as early as possible.

In a ramping interface the user should always be able to obtain more information by going to the next stage, and the action required to get to that stage should be proportional to the amount of information provided in the current stage. It should only require a simple action such as reading a pop-up window for a user to go to early stages. Later stages might require the user to click on an icon, trading off simplicity for increased control of what information is displayed. Note that stages are not defined by display actions taken by the agent, but rather are defined as pieces of information that can be individually processed by a user. For example, a Web page with a large-font title, keywords in boldface, and then the body of the page would contain three stages even if the page is rendered all at once, because a reader can easily process each chunk of information separately and use that information to decide whether to read further.

Of the three implemented JITIR agents, Margin Notes best illustrates the idea of a ramping interface.

The first stage could be referred to as the no-action, no-information stage. In this stage the system analyzes a section of a Web page and decides whether to leave any suggestion at all. There are several reasons a section might not be annotated. First, the best suggestions may still be below the required relevance threshold. The section may also be too small to annotate, or the document as a whole might be too small. At this first stage, the input is simply a passive sensor watching the user's actions. No user action or attention is required to show the agent what to do. The output at this stage is potentially nothing: the agent simply passes the HTML to the browser and continues.

When Margin Notes determines that a suggestion is above a certain relevance threshold, it automatically jumps to the second stage and shows a suggestion note. In keeping with the philosophy that jumping to the next stage should be commensurate with the amount of work or attention required by the current stage, no user thought or action is required to display the suggestion note.

At this point the user may ignore the suggestion entirely, and the interaction has cost nothing more than some screen real estate and a very minor cognitive load. If the user wishes to go to the next stage, he or she can quickly view a row of filled-in circles indicating the relevance value for that suggestion.

The next stage is reached by reading the suggestion description, which requires further attention on the part of the user. Information in the suggestion note is as concise as possible to allow rapid scanning for content. The box also contains many different kinds of information to try to contextualize the suggestion. For example, when e-mail is displayed, the box contains subject, author, date, and archive file name in the hope that at least one of these will be a good indication of what the suggestion is about.

At the fourth stage of Margin Notes the system displays the most relevant keywords. Going to this stage requires physical action by the user (a mouse-over), and gives an incremental increase in the amount of information returned. While keyword information could have been included in the original suggestion (reducing the system to a four-stage ramping interface), it was decided that this information made the suggestion note too cluttered. To jump to the final stage, the user clicks on the link and obtains the fully suggested text. At this point the user is completely committed to seeing the text and has been distracted from the primary task. It is hoped that if the user arrives at this point, the suggested text is worth the attention spent.

Note that the actions required to go through the stages of the Margin Notes ramping interface in order are also the natural actions to obtain more information in the context of Web browsing. The user sees a link, reads it, moves the mouse to the link,

and clicks. This action sequence allows the user to jump to the full suggestion quickly, while still seeing all stages.

When designing a ramping interface, it is also helpful to consider at what stage a user is likely to receive the information needed. In the Margin Notes system, it is assumed that most of the time users will receive information from the full suggested text, but that occasionally they will be reminded by the suggestion note itself and never need to follow the link. In contrast, a JITIR agent intended for an automobile or other attention-demanding environment might be designed such that users normally need not go past the first stage of a suggestion.

Evaluations

Two user studies have been performed. The first is a controlled-task experiment that compares the usefulness of JITIR agents to a traditional search engine. The second study evaluates the information retrieval used by the implemented JITIR agents. It also examines how the database used by an RA affects the quality of suggestion. These systems have also been deployed for several years, and user feedback has been used to improve the systems.

Controlled task evaluation. The first experiment compares JITIR agents to nonproactive information tools such as search engines. Twenty-seven subjects were recruited from the MIT community and were divided randomly into experimental and control groups. Subjects from both groups were asked to write a newspaper-style article about housing at MIT, currently a hotly debated topic around campus. Control group users were given a normal Emacs text editor and a Web browser accessing the MIT *Tech* newspaper search page.[25] Experimental group subjects were given an Emacs augmented by the Remembrance Agent and the MIT *Tech* search page augmented by Margin Notes. The Emacs RA, Margin Notes, and *The Tech* search page all pulled from the same archive of 16 240 news articles. All interactions with the information tools were logged, and subjects were also given pre- and post-task surveys.

A control group of 14 and an experimental group of 13 subjects were tested. Of these, two outliers from the control group viewed a number of articles more than 1.5 times the inner-quartile distance from the third quartile and were eliminated. One of these two reported that he had finished early and browsed *The Tech* about other issues for the remaining time, the other reported she had little knowledge about housing and so wrote about "what *The Tech* says about housing." Results were consistently good for the RA in terms of preference and usage. Margin Notes, however, did poorly in both categories. This outcome is not unexpected, since Margin Notes was never displaying information when the subjects were performing their primary task. Within the setup of this experiment, the amount a subject used the search engine set an upper bound to how many articles would even be displayed using Margin Notes.

Subjects who were given all three tools showed a consistent preference for the RA over the search engine and the search engine over Margin Notes in terms of ranking, overall usefulness, and whether they would want the tool for a similar task. As can be seen in Table 1, 78.5 percent of experimental subjects ranked the RA as their number one choice. (Because some answers were left blank, percentages do not add up to 100 percent.) The RA was also rated around a point (out of seven) higher than the search engine for both usefulness and if the subject would want to use the system for a similar task. Margin Notes, in contrast, was consistently rated lower than the other two systems. The differences between the RA and the rank order of the search engine and whether the system was useful are both statistically significant ($p = 0.05$); the differences between the search engine and Margin Notes and the differences in whether the subject would want to use the system again are not significant. Errors are to the $p = 0.05$ level.

The usage patterns for the different tools point to similar conclusions. As can be seen from Table 2, subjects from the experimental group viewed around three times as many different *Tech* articles as did those in the control group, and within the experimental group subjects viewed around two-and-a-half times as many articles using the RA as they did using the search engine. The difference between total pages viewed in the two groups and the differences between search engine and RA use within the experimental group are significant to the 0.01 level. Less than one-third of the experimental subjects viewed any documents suggested by Margin Notes at all, and even those did not view many.

The essays themselves were also examined and coded to see whether a difference could be found between the control and experimental groups. Articles were blinded and coded for number of facts mentioned, number of references to *The Tech*, and overall quality. However, individual variance was high, and no

statistically significant difference was found between the two groups.

From these results it is clear that the RA is not just an alternative for a search engine; rather it facilitates the retrieval of information that would not otherwise be viewed at all. This conclusion is supported by the fact that search engine use was not significantly reduced when users were given the RA, and the total number of articles viewed increased. This fact follows directly from the accuracy versus effort trade-off described earlier. As one subject commented, "the hits [from the RA] weren't as effective or relevant as the ones from the search engine, but I would never bother using the search engine." Two other experimental subjects commented that they "would be writing opinions, and the RA would bring up the facts to support those opinions automatically." This comment is in contrast to a subject in the control group, who complained that he bothered to check his facts with the search engine, but it was a great deal of effort, and he almost did not bother. Perhaps the best review of the RA was a subject who commented that it was "almost an unfair advantage to have the RA."

A few criticisms were made about the RA that could easily be remedied. First, several subjects commented that they would start receiving very good suggestions, but that the display would "settle" and show the same suggestions after a few paragraphs had been written. This situation occurred because the RA was set up with only one scope, based on the past 500 words of the text being written. As described earlier, the RA is configurable for multiple scopes, each describing a different database or based on a different amount of text. It would have been simple to configure the RA so a third of the suggestions were based on the past 20 words being written, a third based on the past 100 words, and a third based on the past 500 words written. However, this particular experiment only used the one scope.

Several subjects also requested the ability to highlight a particular region or word for which they wanted more information. While the RA does not have this feature yet, it does have the capability to perform full manual queries based on individual strings. However, subjects were not informed about this feature because the experiment was designed to examine the proactive nature of the RA. The highlighting feature will be added in a future release.

Table 1 Experimental group reported preferences (n = 13)

	Search Engine	Emacs RA	Margin Notes
Found useful (1–7)	3.8 ± 1.3	5.1 ± 0.7	2.8 ± 1.1
Would want again (1–7)	4.8 ± 1.1	6.0 ± 0.8	3.9 ± 1.2
Percent ranked #1	27	78.5	9
Percent ranked #2	64	15	9
Percent ranked #3	9	8	82

Table 2 Number of unique *Tech* articles viewed (n = 12, n = 13)

	Total	Search Engine	Emacs RA	Margin Notes
Control mean	2.8 ± 1.7	2.8 ± 1.7		
Control median	2	2		
Experimental mean	7.3 ± 2.4	1.9 ± 1.0	4.9 ± 1.9	0.5 ± 0.5
Experimental median	8	2	6	0

Information retrieval test. A second experiment was performed explicitly to test the query-free information retrieval system used by the RA, Margin Notes, and Jimminy. Media Lab researchers were asked to submit conference papers and articles that they had converted to HTML and placed on the Web. These papers were annotated using Margin Notes, running with a database containing a subset of the INSPEC collection of abstracts and citations. The database contained 152860 citations that had been picked based on IEEE thesaurus labels that matched MIT Media Lab areas of research (e.g., "user interfaces," "computer vision," etc.). Margin Notes was patched to display all annotations regardless of relevance, and the relevance score was blanked out. The researchers were then asked to rate each annotation in terms of relevance to their paper in general, relevance to the section it annotates, and whether the citation would be useful if they were writing or rereading their paper. Nine researchers were asked to evaluate the annotations on their papers, for a total of 112 specific annotations.

Generally, the annotations were rated highly for relevance. As can be seen in Table 3, the average score was 3.3 out of 5 for relevance to the paper in general and 3.4 out of 5 for relevance to a specific section. In all, 7 ± 9 percent of the annotations received a score of four or five for general relevance, and 4 ± 9 percent for relevance to a specific section ($p = 0.05$).

Table 3 INSPEC annotation rating breakdown (5 = best)

	General Relevance	Section Relevance	Usefulness
Score = 1	16%	21%	32%
Score = 2	16%	9%	18%
Score = 3	21%	16%	15%
Score = 4	17%	19%	18%
Score = 5	30%	35%	17%
Average score	3.3	3.4	2.7
Error	0.3	0.3	0.3

Usefulness scores were not as good as relevance scores. The average usefulness score was only 2.7, with 35 percent of the annotations receiving a 4 or 5. With a further look at the data, it is clear that, at least for this task, relevance was a necessary but not sufficient condition for usefulness. Of particular interest is that, of the 4 ± 7 percent of citations that were rated low on usefulness (1 or 2) but high on general relevance (4 or 5), 100 percent were noted as being not useful because the citation was already known by the person doing the rating. Moreover, 68 percent were not only known, but were in fact written by the person doing the rating. This result is encouraging for three reasons. First, it indicates that higher usefulness ratings could be obtained by using a small amount of domain knowledge to avoid suggesting references that the user most likely already knows. For example, in this case the system could avoid listing citations that are already referenced in the paper or written by the user of the system. Second, the fact that these references were relevant but already known by the author of the paper being annotated means the annotations would probably be useful to someone less well-versed in the particular subject. Third, just because a reference is already known does not mean it is not useful. Some of the readers rated known annotations as still being useful, commenting that they were a good reminder.

Because the relevance score is used both as an indication of usefulness and for filtering low-relevance suggestions, it was hoped that the score would have a strong correlation to how relevant a person would find an annotation. Two relevance scores were tested. The first is normalized by the number of unique words in the query (the section being annotated). When this normalization is not performed, relevance scores for long queries can be orders of magnitude higher than scores for short queries. The second was the raw unnormalized score. Unfortunately, the nonnormalized score had only a low correlation to human judgments of relevance ($r = 0.36$, $p = 0.0002$), accounting for only 13 percent of the variation. Normalized scores were even lower ($r = 0.13$) and were not statistically significant. The reason is query length correlates to relevance ($r = 0.27$, $p = 0.05$), because with more words in the query, the text-retrieval system can find a better match. Correlations between scores and usefulness were not significant, presumably because highly relevant annotations were more likely to already be known by a reader and thus not be useful.

Related work

Most directly related to this work are other JITIR agents: systems that proactively provide information based on a person's local context in a nonintrusive yet accessible manner. Several such systems are described below.

Watson[26] is a system that automatically produces and submits AltaVista** or other Web queries based on a user's current Web page or Microsoft Word** document. Results are clustered and displayed in a separate window. Watson is designed both to proactively display information and to augment queries to Web pages with local context. It also includes domain-specific features, such as automatically searching the Arriba Vista image search engine whenever a Microsoft Word user creates an image caption.

SUITOR[27] watches a person's Web browser, word processor, and other applications and provides potentially useful information above the task bar at the bottom of the screen. It is based on a blackboard architecture with multiple agents, each looking for domain-specific information. For example, if a person is looking at the IBM Web page and also looking at financial pages, SUITOR will present IBM stock quotes at regular intervals.

Letizia[28] automatically recommends Web pages a person might want to visit, based on a short-term user profile and the Web page currently being viewed. Letizia automatically creates a user profile by compiling keywords contained in previous pages viewed by the user. It then acts as an "advance scout," following links from the user's current Web page and bringing up pages that match the user's profile. In most JITIR agents the source of suggested information is personalized and static (though slowly updated), and the user's current context is used to retrieve

this information. Letizia does just the opposite: the source of information is the Web pages, and documents are suggested based on the user's profile that changes over the course of hours or days. Letizia also shows the entirety of a recommended Web page in a separate window rather than presenting a summary of potential hits and letting the user pick which suggestions to view.

Kenjin**[29] is a brand new commercial product by Autonomy that, according to user guides and press releases, provides information from product databases, Web searches, or the user's personal computer that are related to information in currently running applications. Being a commercial product and having just been released, few details are available.

Finally, RADAR[11] is a different front end for the Remembrance Agent that uses Microsoft Word instead of Emacs and displays suggestions in a separate window. RADAR was built in collaboration with the JITIR work described here, and also uses Savant as the information-retrieval engine.

There are also several JITIR agents that proactively provide information based on local context but are domain-specific.

Fixit[30] is an expert system for copy machine repair that automatically brings up manual pages relating to the fault being diagnosed. It uses the table of contents for the repair manual to find useful pages, based on the diagnosis produced by the expert system. Although the techniques used were domain-dependent, the corpus (manual pages) was still a legacy system and did not need to be hand-coded for the system.

WebWatcher[31] is similar to Letizia, highlighting hyperlinks that best match a user's stated interest. As with Letizia, recommendations are only pulled from links on the page currently being viewed, based on a user profile. However, in WebWatcher, recommended links are indicated by surrounding the link with a special icon. The system gives no further information about the link or why one might want to follow it. WebWatcher uses the topology of links between Web pages to make recommendations and is therefore only applicable to the Web.

The Peace, Love, and Understanding Machine (PLUM) system[32] will automatically add hyperlinks to disaster news stories to better give a reader a personalized understanding of the news. For example, a report that 220 people were killed in an earthquake in a small town in Japan would contain hyperlinks from the number "220" to a description of what percentage of the population was killed and how many people that percentage would represent in the reader's own hometown.

Flyswat**[33] is another recent commercial venture. Flyswat operates as a plug-in to a Web browser and highlights words or phrases it identifies in a database. Clicking on the highlight can lead to dictionary or thesaurus lookups or e-commerce pages that will sell products related to the word.

JITIR agents also share similarities with many personalized information systems. Of these, the most relevant are Lifestreams,[34] the Forget-Me-Not system,[35–37] and Lotus Agenda**.[38] All these systems are designed to organize personal information with minimal user intervention. However, Forget-Me-Not is a direct memory aid, allowing users to search a database of their own automatically recorded past actions. It falls into the same category as search engines and other interactive information retrieval systems and is not designed to be proactive. Lifestreams is an organizational structure to personal data rather than a JITIR agent, though it does provide a notification system based on time. Agenda is a database program that includes contextual triggers that automatically perform an action when data that meet a particular criterion are entered.

JITIR agents are also related to work in the area of context-aware applications, especially those that use wearable computers. One such system is Audio Aura,[39] an audio-based wearable computer that uses ambient sound to automatically indicate e-mail, group activity, and location-based information. Nomadic Radio[40] is another audio-based wearable system to deliver news, voice mail, and e-mail. In particular, Nomadic Radio uses a concept called "dynamic scaling" that is quite similar to a ramping interface. A third related system is the Context-Aware Archaeological Assistant,[41] by which location sensed by a GPS is used to provide field notes about animals in the area. This work is also related to many efforts in augmented reality[42,43] and ubiquitous computing.[44]

Finally, several interface techniques have been developed for presentation of information related to current text, either automatically or by an author's design. One such system is VOIR,[45] which combines elements of a document with a user's interest profile to automatically create hyperlinks. Another is

the XLibris pen-based system,[46] which supports the ability to hand-annotate documents using margin links. Similarly, Fluid Links[47] allows Web designers to create "glosses," which are a way of presenting information incrementally.

Conclusions

In conclusion, just-in-time information retrieval agents are systems that proactively provide information based on a person's local context in a general, task-independent way. They are related to search engines, alarms, and news-clipping software, but differ from each.

The Remembrance Agent, Margin Notes, and Jimminy have all been in use in the field for over three years. Although controlled experiments are useful for validating theories, it is these experiments "in the wild" that give the best indication of what those theories should be. Reports from long-term users have verified the assertion that JITIR agents are not simply a substitute for a traditional search tool; using a JITIR agent changes how people use information in ways that having a search engine alone does not. For example, the comments of the two controlled-experiment subjects that "I typed opinions, and the RA gave me the facts to support them automatically" have been repeated many times in conversations with long-term users. Variations include users who answer more e-mailed questions than they normally would because the RA makes it simple, and users becoming generally more aware of the historical context surrounding issues about which they are reading or writing.

It is also clear from long-term use that JITIR agents are situational applications: their effectiveness depends a great deal on the environment in which they are used. If the environment and the agent are mismatched, for example, using the INSPEC database when writing e-mail about nonresearch, then the results are not useful. Similarly, a database of personal e-mail works well for the owner of that e-mail but does not produce results as good when used by another. These observations are the inspiration for designing techniques that are generally applicable, so the agents can be adapted quickly to different domains, corpora, and individual preferences. It is also the inspiration for producing interfaces that are ignorable, so the inevitable false-positive suggestions do not become more of a distraction than their occasional usefulness is worth.

Acknowledgments

We would like to thank our sponsors at British Telecom and Merrill Lynch. We would also like to thank Thad Starner, without whom the project would never have been started, and the RA team of Jesse Rabek, Alex Park, Jesse Koontz, Pammie Mukerji, Matthew Burnside, Aneel Nazareth, Bayard Wenzel, and Jan Nelson, without whom the project would never have been completed. Finally, we would like to thank the reviewers of the papers that have been written on this subject over the years, and the Remembrance Agent for bringing those reviewer comments to the foreground just when they were needed the most.

**Trademark or registered trademark of Yahoo! Inc., Microsoft Corporation, The Open Group, Linus Torvalds, Reflection Technology, Inc., Handykey Corporation, Insilco Corporation, AltaVista Company, Autonomy, Inc., Flyswat, Inc., or Lotus Development Corporation.

Cited references

1. B. Rhodes and T. Starner, "The Remembrance Agent: A Continuously Running Information Retrieval System," *The Proceedings of the First International Conference on Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, London (April 1996), pp. 486–495.
2. B. Rhodes, "Margin Notes: Building a Contextually Aware Associative Memory," *Proceedings of the International Conference on Intelligent User Interfaces (IUI'00)*, New Orleans, LA (January 9–12, 2000), pp. 219–224.
3. B. Rhodes, "The Wearable Remembrance Agent: A System for Augmented Memory," *Personal Technologies: Special Issue on Wearable Computing* **1**, 218–224 (1997).
4. T. Starner, *Lizzy: MIT's Wearable Computer Design 2.0.5*, http://www.media.mit.edu/wearables/lizzy/ (1997).
5. B. Rhodes, "Wearable Computing Meets Ubiquitous Computing: Reaping the Best of Both Worlds," *The Third International Symposium on Wearable Computers (ISWC'99)*, San Francisco, CA (October 18–19, 1999), pp. 141–149.
6. T. Starner, D. Kirsch, and S. Assefa, "The Locust Swarm: An Environmentally-Powered, Networkless Location and Messaging System," *First International Symposium on Wearable Computers (ISWC'97)*, Cambridge, MA (October 13–14, 1997), pp. 169–170.
7. R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems* **10**, No. 1, 91–102 (January 1992).
8. N. Minar, M. Gray, O. Roup, R. Krikorian, and P. Maes, "Hive: Distributed Agents for Networking Things," *Proceedings of the First International Symposium on Agent Systems and Applications and Third International Symposium on Mobile Agents (ASA/MA'99)* (1999), pp. 118–129.
9. S. E. Johnson, P. Jourlin, K. Spark Jones, and P. C. Woodland, "Spoken Document Retrieval for TREC-8 at Cambridge University," to appear in *NIST Special Publication: The Eighth Text REtrieval Conference (TREC 8)* NIST, Gaithersburg, MD (2000).
10. A. Pentland, B. Moghaddam, and T. Starner, "View-Based and Modular Eigenspaces for Face Recognition," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA (June 1994), pp. 21–23.

11. I. B. Crabtree, S. Soltysiak, and M. Thint, "Adaptive Personal Agents," *Personal Technologies* **2**, No. 3, 141–151 (1998).
12. S. Robertson, S. Walker, and M. Beaulieu, "Okapi at TREC-7: Automatic *ad hoc*, Filtering, VLC and Interactive," *NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC 7)*, NIST, Gaithersburg, MD (1999), pp. 253–264.
13. B. Rhodes, *Just-in-Time Information Retrieval*, Ph.D. dissertation, MIT, Media Arts and Sciences, Cambridge, MA (June 2000).
14. G. K. Zipf, *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*, Addison-Wesley Publishing Co., Reading, MA (1949).
15. J. Payne, J. Bettman, and E. Johnson, *The Adaptive Decision Maker*, Cambridge University Press, Cambridge, UK (1993).
16. S. Jarvenpaa, "The Effect of Task Demands and Graphical Format on Information Processing Strategies," *Management Science* **35**, 285–303 (1989).
17. J. Russo, "The Value of Unit Price Information," *Journal of Marketing Research* **14**, 193–201 (1977).
18. R. Miller, "Response Time in Man-Computer Conversational Transactions," *AFIPS Conference Proceedings of the Fall Joint Computer Conference* **33**, Part 1 (1968), pp. 267–277.
19. B. Jansen, A. Spink, and J. Bateman, "Searchers, the Subjects They Search, and Sufficiency: A Study of a Large Sample of EXCITE Searches," *Proceedings of WebNet-98* (1998).
20. D. Harman, "Overview of the Third Text REtrieval Conference (TREC-3)," *NIST Special Publication 500-226: Overview of the Third Text REtrieval Conference (TREC-3)*, NIST, Gaithersburg, MD (1995), pp. 1–20.
21. D. Hull, "The TREC-7 Filtering Track: Description and Analysis," *NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC 7)*, NIST, Gaithersburg, MD (1998), pp. 33–56.
22. C. D. Wickens, *Engineering Psychology and Human Performance*, *Second Edition*, Harper Collins Publishers, New York (1992).
23. Wickens, *op. cit*., p. 98.
24. Y. Miyata and D. Norman, "Psychological Issues in Support of Multiple Activities," *User Centered System Design*, D. Norman and S. Draper, Editors, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ (1986).
25. *The Tech* Archive Search, see http://www-tech.mit.edu/search.html.
26. J. Budzik and K. Hammond, "Watson: Anticipating and Contextualizing Information Needs," *Proceedings of the Sixty-Second Annual Meeting of the American Society for Information Science* (1999).
27. P. Maglio, R. Barrett, C. Campbell, and T. Selker, "SUITOR: An Attentive Information System," *The Proceedings of the International Conference on Intelligent User Interfaces (IUI'00)*, Henry Lieberman, Editor, ACM Press (January 9–12, 2000), pp. 169–176.
28. H. Lieberman, "Autonomous Interface Agents," *Proceedings of CHI'97*, Atlanta, GA (March 1997), pp. 67–74.
29. Autonomy, Inc., San Francisco, CA, see http://www.kenjin.com.
30. P. Hart and J. Graham, "Query-Free Information Retrieval," *IEEE Expert/Intelligent Systems & Their Applications* **12**, No. 5, 32–37 (September, October 1997).
31. T. Joachims, D. Freitag, and T. Mitchell, "WebWatcher: A Tour Guide for the World Wide Web," *International Joint Conference on Artificial Intelligence* (August 1997), pp. 770–775.
32. S. Elo, *PLUM: Contextualizing News for Communities Through Augmentation*, master's thesis, MIT, Media Arts and Sciences, Cambridge, MA (1995).
33. Flyswat, Inc., see http://www.flyswat.com.
34. E. Freeman and D. Gelernter, "Lifestreams: A Storage Model for Personal Data," *ACM SIGMOD Bulletin*, 80–86 (March 1996).
35. M. Lamming, P. Brown, K. Carter, M. Eldridge, M. Flynn, and G. Louie, "The Design of a Human Memory Prosthesis," *The Computer Journal* **37**, No. 3, 153–163 (1994).
36. M. Lamming and M. Flynn, "Forget-Me-Not: Intimate Computing in Support of Human Memory," *Friend21: International Symposium on Next Generation Human Interface*, Meguro Gajoen, Japan (1994), pp. 125–128.
37. P. Brown, J. Bovey, and X. Chen, "Context-Aware Applications: From the Laboratory to the Marketplace," *IEEE Personal Communications* **4**, No. 5, 58–63 (October 1997).
38. S. Kaplan, M. Kapor, E. Belove, R. Landsman, and T. Drake, "Agenda: A Personal Information Manager," *Communications of the ACM* **33**, No. 7, 105–116 (July 1990).
39. E. Mynatt, M. Back, and R. Want, "Designing Audio Aura," *Proceedings of the Conference on Human Factors in Computing Systems (CHI'98)*, ACM Press (1998), pp. 566–573.
40. N. Sawhney and C. Schmandt, "Nomadic Radio: Scalable and Contextual Notification for Wearable Audio Messaging," *Proceedings of the ACM SICHI Conference on Human Factors in Computing Systems*, Pittsburgh, PA (May 15–20, 1999), pp. 96–103.
41. N. Ryan, J. Pascoe, and D. Morse, "Enhanced Reality Fieldwork: The Context-Aware Archaeological Assistant," V. Gaffney, M. van Leusen, and S. Exxon, Editors, *Computer Applications in Archaeology* (1997, 1998).
42. J. Rekimoto, Y. Ayatsuka, and K. Hayashi, "Augmentable Reality: Situated Communications Through Physical and Digital Spaces," *Digest of Papers: The Second International Symposium on Wearable Computers*, Pittsburgh, PA, IEEE Computer Society (October 19–20, 1998), pp. 68–75.
43. S. Feiner, B. MacIntyre, and D. Seligmann, "Knowledge-Based Augmented Reality," *Communications of the ACM* **36**, No. 7, 52–62 (July 1993).
44. M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," *Communications of the ACM* **36**, No. 7, 75–84 (July 1993).
45. G. Golovchinsky, "What the Query Told the Link: The Integration of Hypertext and Information Retrieval," *The Proceedings of HyperText'97*, Southampton, UK, ACM Press (1997), pp. 67–74.
46. M. Price, G. Golovchinsky, and B. Schilit, "Linking by Inking: Trailblazing in a Paper-Like Hypertext," *The Proceedings of HyperText'98*, Pittsburgh, PA, ACM Press (June 1998), pp. 30–39.
47. P. Zellweger, B. Chang, and J. Mackinlay, "Fluid Links for Informed and Incremental Link Transitions," *The Proceedings of HyperText'98*, Pittsburgh, PA, ACM Press (June 1998), pp. 50–57.

Accepted for publication April 13, 2000.

Bradley J. Rhodes *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: rhodes@media.mit.edu).* Dr. Rhodes is a recent graduate of MIT's Media Lab, where he worked under the direction of Pattie Maes in the Software Agents Group. His research areas included the study of intelligence augmentation, just-in-time information retrieval, ubiquitous computing, context-aware applications, and wearable

computing. He is one of the "Media Lab Cyborgs" and for the past four years has been an active member of the MIT Wearable Computing Project. He received his Ph.D. degree from the MIT Media Lab in 2000, his S.M. degree from the MIT Media Lab in 1996, and an S.B. degree in computer science from MIT in 1992.

Pattie Maes *MIT Media Laboratory, 20 Ames Street, Cambridge, Massachusetts 02139-4307 (electronic mail: pattie@media.mit.edu).*
Dr. Maes is an associate professor at MIT's Media Lab, where she founded and directs the Software Agents Group and is principal investigator of the Media Lab e-markets Special Interest Group. Her team's work has included personalized information filtering agents, agents that automate behavior patterns, matchmaking agents, just-in-time information retrieval agents, and agents that buy and sell on behalf of a user. Dr. Maes received her doctoral degree in computer science and artificial intelligence from the Vrije Universiteit Brussel in Belgium in 1987.