Summarizing technical support documents for search: Expert and user studies

by C. G. Wolf
S. R. Alpert
J. G. Vergo
L. Kozakov
Y. Doganata

One factor that may affect whether users of technical support Web sites can rapidly find information relevant to their needs is the quality of the summary of documents returned as the result of search queries. This paper reports on two studies that were part of an effort to create high-quality machine-generated summaries for the presentation of search results for technical support documents. The initial study asked experts to compose document summaries. The results of the first study were used to guide the development of heuristics for generating programmatic summaries that were tested in the second study, which was a user evaluation that compared the effectiveness of four types of document summaries for search purposes: programmatic summaries based on selective sentence extraction using knowledge of the semantic structure of documents, a term-hits-in-context (THIC) summary, the current summaries on the company's live site, and document titles alone. This comparison sought to determine the techniques most likely to help users find information, hence increasing customer goal attainment and satisfaction. The implications of our results for summarizing technical support documents for search are discussed.

In the not so distant past when customers needed technical support, they would pick up the telephone and call an expert at a help desk. However, the cost of providing support to customers, partners, and employees has placed an increasing burden on corporate profitability. The pressure to reduce costs combined with the rapid growth of the Web has caused companies to move support from human experts to the Web. Indeed, many companies charge for the formerly free telephone-based human service, motivating customers to seek assistance on-line. As Ehrlich and Cash[1] have noted, the skills of a help desk organization may be hard to replicate with on-line self-service, but increased human support costs will force many companies to look at cost-effective solutions implemented through the Web.

Coupled with the need to reduce costs is the often conflicting goal of increasing customer satisfaction. The quality of support and service can be the differentiator that sets one business apart from its competitors. When customers go to the Web site of a business seeking information or the solution to a problem, they expect to find it quickly and with a minimum of effort. If they fail, they may turn to another vendor. Thus, it is important for a business to develop and maintain customer loyalty by ensuring that customers achieve their goals when they search for information on its technical support Web site.

There are many aspects of providing high-quality technical support on the Web. In addition to the obvious need to provide high-quality documents that address users' support needs, there is a requirement

0018-8670/04/$5.00 © 2004 IBM

for high-quality search facilities that allow users to find the documents related to a specific information need. One factor that may affect whether customers are able to find the information they need is the quality of the summaries of documents retrieved as the result of a search. (A Web search query typically results in a *hitlist* [a list of documents displayed on the screen that match the search criteria specified by the user]; each item in that hitlist represents a document or page and usually includes the document's title and a summary of the document.) This is particularly important when many results are retrieved for a query. In this case, users must rely on the titles and summaries to determine which documents are relevant.

A large business may have tens or hundreds of thousands of support documents, typically written by different authors and often not necessarily originally intended for customers. Perhaps the best approach to document summarization from a readability and usability perspective would be to have a cadre of human abstractors handcraft summaries for each document. However, creating effective summaries manually is an expensive proposal, especially for organizations that have huge legacy repositories containing documents without such abstracts. In particular, the company Web site we studied has a large repository containing numerous documents without any summaries and a small number of documents with summaries that we and others judged to be inadequate for customer needs. Thus, generating summaries programmatically is a more cost-effective solution in this case.

There are a number of approaches to programmatically generating summaries of documents. We address the method or style that is most effective from a user's perspective for technical support documents on a company's Web site. This paper reports on an empirical user evaluation of different types of summaries for the presentation of search query results to be used on the technical support Web site of a large computer manufacturer. The purpose of this evaluation is to determine the types of summarization techniques that are more likely to help customers find information accurately and quickly and, hence, increase both customer goal attainment and customer satisfaction. This study of the effectiveness of different summary types was part of a larger effort to improve the overall user experience on the technical support Web site.

Search engines on the Web have several different approaches to summary creation. Some of the major Web search gateways, such as Google**, AllTheWeb**, and AltaVista** find snippets of text that contain user search terms in each document and display these snippets to form the document summary. In the summary, user search terms appear highlighted. This method, sometimes called *term hits in context* (THIC), is also used by some corporate technical support search sites, such as those of Accenture[2] and Apple Computer, Inc.[3]

Some enterprise sites do have human-authored document summaries. Such summaries are typically incorporated into documents as meta-data in the HTML (HyperText Markup Language) encoding of a document, for example in the document's *description* metatag. The search engines on these sites use this meta-data as a document's summary. Other search engines simply use the first *n* characters or words of the body of a page as the page's summary, while still others use a hybrid approach combining several of these techniques; for example, a document's summary is obtained from the description (or otherwise named) meta-data field, but if that is not present in the document, the first 255 characters of the page are used. Google combines handcrafted summaries (if available) with THIC summaries.

Another method of generating summaries of documents programmatically is known as *sentence extraction.*[4,5] With this method, complete sentences are taken from the document to compose the summary (this type of summary is sometimes referred to as an *extract* rather than an abstract or simply a summary). The algorithms for sentence selection may take into account lexical content, the position of a sentence in the document, neighboring sentences, headings indicated by markup tags or layout, as well as other factors. Other summarization techniques seek to apply some of the same transformations as human professional summarizers do, such as sentence reduction and combination.[6] (Additional information on summarization algorithms, techniques, and projects can be found in References 7-10.)

We had at our disposal a programmatic sentence extraction text summarizer.[4] Before electing to apply this tool to the technical support documents in the corpus we were studying, we decided we should understand how experts composed summaries. There are several reasons why we wanted to see how experts summarized the technical support documents, rather than simply applying an existing sentence extraction summarizer to entire documents. First, the corpus to which our sentence extraction summarizer and most others had been tuned is news stories. Technical support documents differ from news stories not only in content, but also in style and structure. In addition, Web support documents may contain images, links, and other non-text elements. Second, when users come to a technical support Web site, they often have a specific problem to be solved. Thus, their task is different from that of a person browsing news stories or searching the Web for information on a topic.

The corpus and tasks of the present study also are different from those of general purpose Web search engines such as Google. Such search engines must compose summaries for the whole gamut of Web documents, encompassing a huge variety of content, style, and structure. The user tasks that these search engines must deal with are limited only by human imagination. Moreover, the structure of such documents cannot be known beforehand. We cannot assume that THIC summaries used by general purpose search engines are the best choice for technical support documents.

The first part of this paper reports on a study in which professional authors of various document types were asked to compose summaries of technical support documents by sentence extraction. The authors were told to create the summaries for the purpose of displaying the results of a search query. The findings from this study were used to guide the heuristics we used for creating the sentence-extraction summaries tested in the subsequent user study.

In the rest of the paper, we describe the Web site and its support documents, the author study that was conducted to learn how experts compose summaries by sentence extraction, the summary conditions tested, including a high-level description of the algorithms used to generate summaries, the design and methods of the study, and its results. We conclude with a discussion of the findings and the resulting implications for the creation of technical support document summaries on the Web.

Web site and documents

Documents used in this study were actual pages from a technical support Web site that is a portion, or subsite, of a large computer manufacturer's enterprise site. The support subsite provides technical support for a large range of hardware and software products, including laptop and desktop computers, servers, point-of-sale systems, computer accessories, application software, developer tools, and so on. Users of the technical support site include both professional information technology specialists and home and home office users.

There are about 500 000 documents on the site, ranging in complexity from highly technical to basic information. The major challenge in deriving heuristics for summarizing these documents is that they are by no means homogeneous; rather, there are many different document types. Documents of different types and even documents of one particular type may be created by different organizations within the enterprise, sometimes with differing authoring tools. The result is a lack of consistency across and within document types with regard to document content, subcomponent structure, and format. There are about 16 major document types (and a number of format variants within many types). The types included download pages, frequently asked questions (FAQs), white papers, general product information, parts information, previously solved problem reports, solutions, and hints.

Figure 1 shows the sections, as indicated by heading names, contained in the main body of several document types in the studied corpus.

Author study

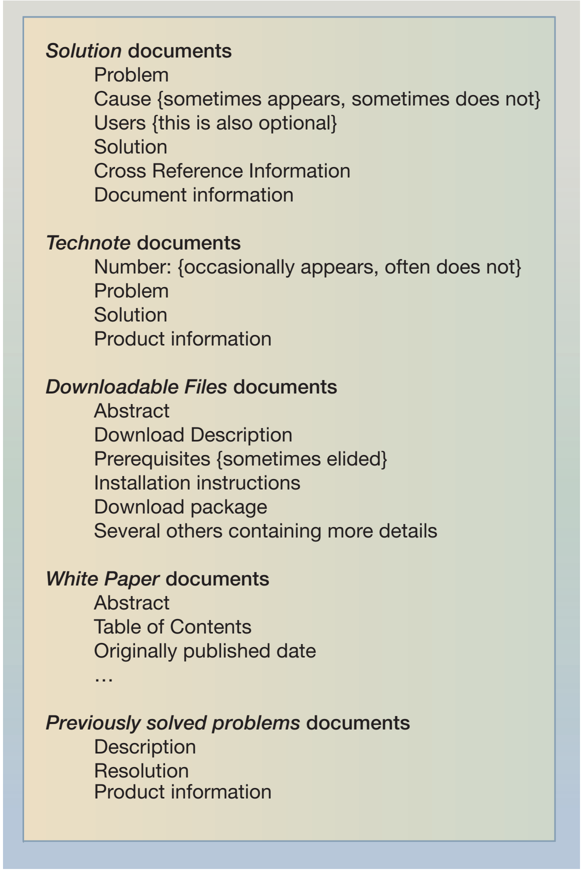
Before running the main summarization comparison study, the research team performed a preliminary study involving expert document authors. The purpose of this initial study was to understand how experts would compose summaries of the site's documents by sentence extraction in order to guide the heuristics for our programmatically generated abstracts. The participants were nine authors of live-site documents of various types. For the study, each author composed summaries for only those document types he or she had authored in the past. Thus, they were familiar with the format and contents of the documents. We told the authors to think of a user searching the technical support Web site and asked them to compose summaries that would be useful for deciding which documents were relevant to the user's information needs.

The authors then composed summaries for sample documents that we provided. They were instructed to construct summaries using whole sentences from the documents. We also informed them that they could optionally format the summary text using spaces, boldface or italic fonts, or other formatting features. The study team then analyzed these expert-authored summaries to glean sentence-extraction-oriented summarization rules applicable to the respective document types.

The mean number of sentences in the author-generated summaries was 2.19 with a standard deviation of 1.59. The sentences chosen were generally from one or at most two sections of the document and tended to be from paragraphs early in those sections. For example, representing the simplest case of selecting an appropriate portion or portions of a document to comprise a summary, some document types contain a section named *Abstract*. This typically contains an introductory summary of the document's contents and, importantly, is crafted by the (human) document author. We had initially decided that for those document types which include an Abstract section, we would choose the entire contents of this section as the document's summary. Based on the author study results, we refined our initial decision as follows: we would use the Abstract section as is, unless that section was too long (a threshold of 255 characters was used), in which case the programmatic sentence extraction summarizer was used to provide an extract of this section. We further generalized this same heuristic to all summaries—if a particular preferred section was too long, we invoked the programmatic sentence extraction summarizer to condense that section.

There are several document types on the support site that are strictly for solving problems encountered with a piece of hardware or software (as opposed to documents with general product information). The structure of these documents typically includes a section labeled *Problem* or *Question* or *Description* (depending on the document type), followed by a section named *Solution* or *Answer* or *Resolution* (there are many minor variations on this theme in these document types and their variants). For such troubleshooting document types, in 81 percent (17 of 21) of the documents used in the study, the expert authors used sentences from the Problem/Question/Description section only. They explained this approach by pointing out that users would be searching for a description of symptoms matching their problems. In other words, the summaries were task-oriented; they depended on the task for which the document was written. In cases where the obvious section was deemed inadequate (mostly in terms of length), the authors took additional text from another section. They were nonetheless careful not to select just any additional information from the document, but rather chose information from specific sections that supported the task-oriented nature of the summary and reflected the task-oriented nature of the documents. For example, in Figure 2, the summary contains all of the short Problem section and the initial sentences of the Solution section. This approach coincided with our own intuitions regarding summaries for these troubleshooting documents.

Figure 1 Main headings in technical support documents of various types in the corpus studied



Solution documents
Problem
Cause {sometimes appears, sometimes does not}
Users {this is also optional}
Solution
Cross Reference Information
Document information

Technote documents
Number: {occasionally appears, often does not}
Problem
Solution
Product information

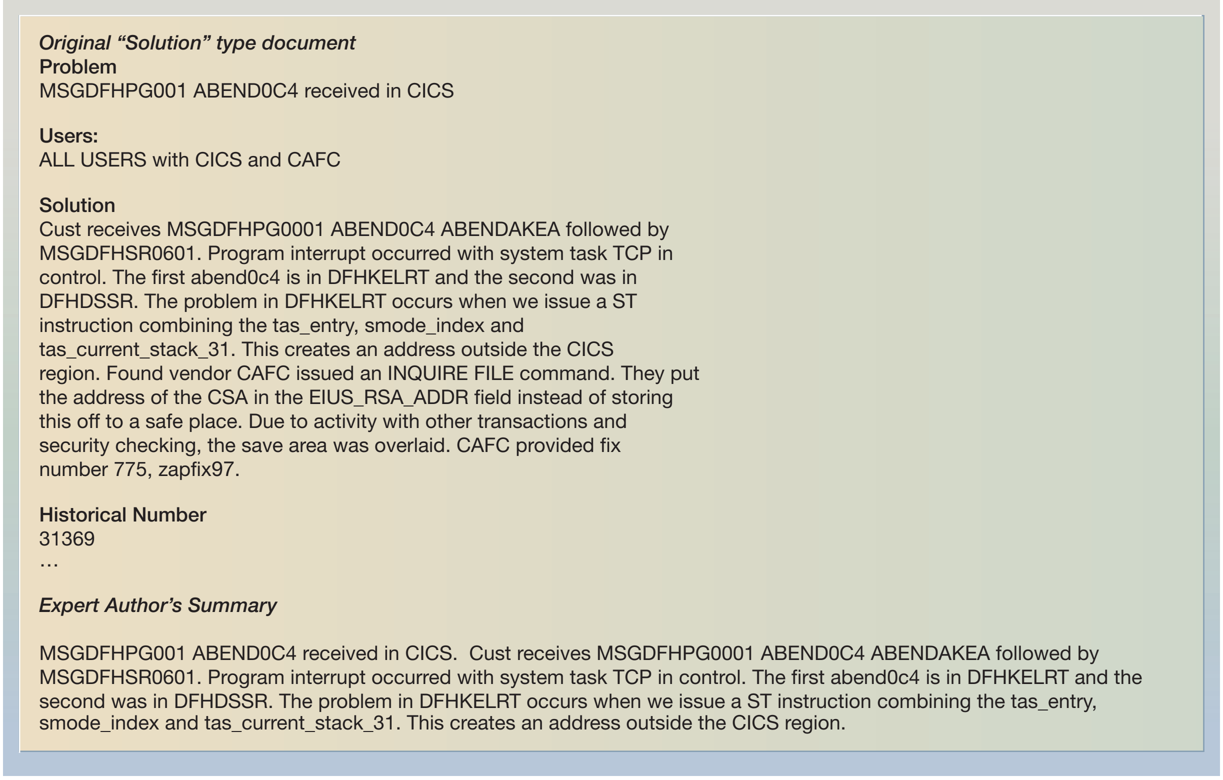
Downloadable Files documents
Abstract
Download Description
Prerequisites {sometimes elided}
Installation instructions
Download package
Several others containing more details

White Paper documents
Abstract
Table of Contents
Originally published date
…

Previously solved problems documents
Description
Resolution
Product information

The authors followed this strategy more generally. For document types with general information or a range of information, they picked sentences from the section that best described the document contents. In some cases, when the relevant section was too brief but contained a link to a PDF (Portable Document Format) file, they took relevant sentences from the linked file. The authors also used boldface and italic

Figure 2 Example of Solution document and its summary as composed by a participant in the document-author study



Original "Solution" type document
Problem
MSGDFHPG001 ABEND0C4 received in CICS

Users:
ALL USERS with CICS and CAFC

Solution
Cust receives MSGDFHPG0001 ABEND0C4 ABENDAKEA followed by MSGDFHSR0601. Program interrupt occurred with system task TCP in control. The first abend0c4 is in DFHKELRT and the second was in DFHDSSR. The problem in DFHKELRT occurs when we issue a ST instruction combining the tas_entry, smode_index and tas_current_stack_31. This creates an address outside the CICS region. Found vendor CAFC issued an INQUIRE FILE command. They put the address of the CSA in the EIUS_RSA_ADDR field instead of storing this off to a safe place. Due to activity with other transactions and security checking, the save area was overlaid. CAFC provided fix number 775, zapfix97.

Historical Number
31369
...

Expert Author's Summary

MSGDFHPG001 ABEND0C4 received in CICS. Cust receives MSGDFHPG0001 ABEND0C4 ABENDAKEA followed by MSGDFHSR0601. Program interrupt occurred with system task TCP in control. The first abend0c4 is in DFHKELRT and the second was in DFHDSSR. The problem in DFHKELRT occurs when we issue a ST instruction combining the tas_entry, smode_index and tas_current_stack_31. This creates an address outside the CICS region.

fonts to emphasize text, or white space to set things apart, and added punctuation to delimit headings or sentences.

Summary types investigated in user study

The main study reported here was a user study in which different document summary types were tested comparatively. We tested four document summary types: the summaries currently shown on the live technical support site, our own programmatically generated abstracts, a terms-in-context (THIC) summary, and no summary at all (that is, titles only). In the rest of this paper, these will be identified as *Live Site*, *Abstract*, *THIC*, and *Titles Only*.

Live Site summary. Live Site summaries were obtained by programmatically performing actual searches via the live-site search engine and caching the results. Interestingly, the summaries on the live site exhibit a broad range of quality, content, and length. In some documents, the author has included handcrafted summary information in a description metatag; when this information exists, it is displayed as the document's summary. These are at times quite succinct. In many cases, the description tag information simply reiterates all or part of a particular section of the visible portion of the document (e.g., a section of the document titled Abstract). In this case the length may range up to over 500 characters (in fact the description field may contain a longer string of text, but the live-site engine truncates this text for the summary; the maximum length used for the summary appears to be in the 500+ character range). For some documents, the Live Site summaries consist merely of a reiteration of the title of the document. Many Live Site summaries consist of the first 255 characters of the body of the document. When the summary is blindly composed of the document's initial 255 characters, it often contains sections or information that might be less useful in a hitlist summary than other more centrally relevant portions of the document. In fact, for some documents, the first 255 characters contain canned introductory document text that gives no clue to a document's distinguishing contents.

Abstract summary. The process for creating the programmatically generated abstracts was a bit more involved. We intended our abstracts to be composed using text from the documents themselves. As mentioned earlier, we did have the option of employing an intelligent programmatic sentence extraction summarization tool, but found that it performed better for purposes of technical document summaries when analyzing a particular section of a document than when attempting to summarize the entire document contents. The question, of course, is how to choose the sections to be summarized for each document type. The study team analyzed each document type and its subcomponent sections and multiple variants to determine which pieces would best be used for a good, cogent summary. Then, as described above, we also investigated how document authors compose summaries by sentence extraction for use in search-result hitlists. The findings from this study were used to help guide the creation of heuristics for creating summaries by sentence extraction. These two sets of analyses led to a set of imperative and conditional heuristics for the composition of abstracts for each document type.

We eventually derived and implemented a set of heuristics for the composition of Abstract summaries for all 16 document types and their numerous variations. The heuristics give primacy to selecting specific sections of the documents and then, conditionally, invoking the sentence extractor tool on contents of specific sections. We also decided that all of our abstracts would begin with the document type (in square brackets), and conclude with the information provided by the *Product Information* or *Document Information* section (depending on document type) that appears in each of the support site documents, all appropriately formatted.

Let us consider some concrete examples. For some document types the summarization based on structure was quite straightforward. For example, *Flash* documents had the following structure, where the section names listed are the literal names in the document unless they appear within angle brackets:

- ⟨Title⟩
- Flashes (literal document type indicator)
- Abstract
- Content

Our summarization heuristics for Flash documents were simply the following. The notation *SentenceExtract* (⟨section-name⟩) means get all the text of the section labeled ⟨section-name⟩, and if its length is less than 256 characters, use it as is; otherwise, invoke the programmatic sentence extraction summarizer to obtain an extract of that text.

1. output "[Flash]"
2. output *SentenceExtract* (Abstract)
3. output the "Product Information" section (with predetermined product-information format defined elsewhere)

On the other hand, some structure-based abstract heuristics were somewhat more complex. For example, although all FAQ documents share the same document type in name, there exist many different formats for such documents. The following describes the process for generating the abstract for FAQ documents. First, FAQ documents on the site were composed using four variant formats. The first variant had the following basic format:

- ⟨Title⟩
- Frequently asked questions (literal document type indicator)
- Question
- Answer
- Cross-reference information (optional—appears in some but not all documents)
  - ⟨Information contained in a table⟩

The second, minor, variant differed from the first only in the presence of colons after the Question and Answer headings (i.e., those subsections were labeled "Question:" and "Answer:"). The format of the third variant was:

- ⟨Title⟩
- Frequently asked questions
- Question (in this variant the content of this Question section is always exactly the same text as the title)
- Answer
  - Question: (with colon)
  - Answer: (with colon)

The format of the fourth variant was essentially the same as that of White Paper documents found on the site, except for the document type indicator:

- ⟨Title⟩

- Frequently asked questions (document type indicator)
- Number:
- Description
- ⟨Idiosyncratic section names specific to the document's content⟩ (optional)
- Related documents (optional)

When we encounter an FAQ document of this last format, we simply wish to invoke the White Paper summarization heuristics. Otherwise, we want the content of the most informative Question section because that should directly speak to the task-based needs of a user for whom this document will be most useful. The resultant heuristics we used to summarize FAQ documents are as follows.

```
1. output "[FAQ]"
2. if there exists a section named "Number:" then abort
   this process and summarize the rest of this document
   using the heuristics for White Paper documents
3. else get the text of the top-level section whose name
   begins with "Question" (handle the cases where the
   section is labeled "Question" or "Question:")
4. if the textual content of the section obtained in step
   3 is the same as the document title, then
   a. output SentenceExtract (⟨the "Question:" (with
      colon) subsection that is subordinate to the top-
      level "Answer" section⟩)
   b. if the length of the text output in step (a) is less
      than the minimum-output threshold, output
      SentenceExtract (⟨the "Answer:" (with colon)
      subsection that is subordinate to the top-level
      "Answer" section⟩)
5. else
   a. output SentenceExtract (⟨the top-level
      "Question/Question:" section's text (as ob-
      tained in step 3)⟩)
   b. if the length of the text output in step (a) is less
      than the minimum-output threshold, output
      SentenceExtract (⟨the top-level section whose
      name begins with text "Answer"⟩)
6. output the "Product Information" section (with pre-
   determined product-information format defined
   elsewhere)
```

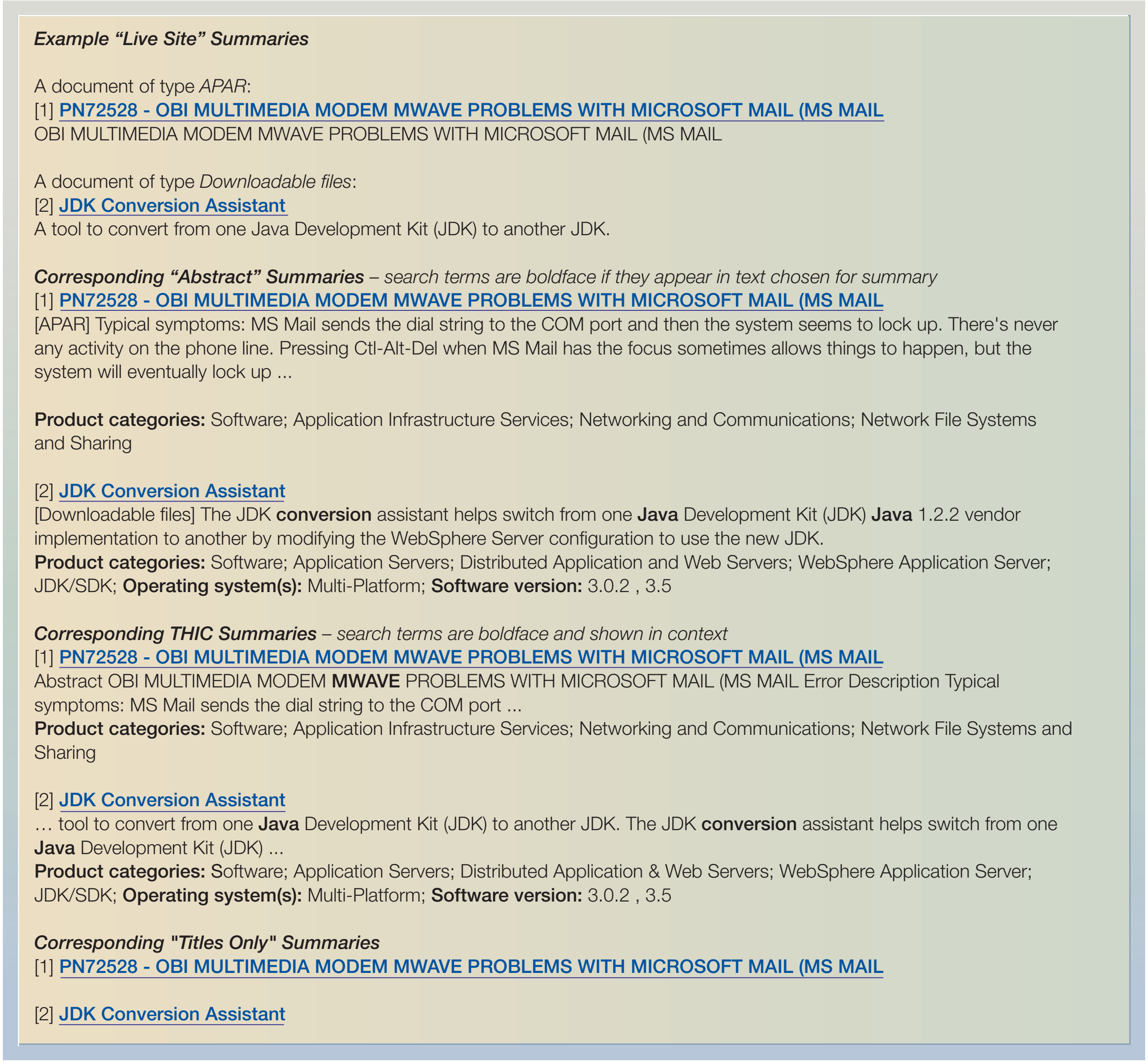
Based on expert authors' summaries and our own analyses, we also applied formatting mechanisms (boldface, spacing) to make our programmatically generated summaries more readable. The examples in Figure 3 demonstrate such highlighting features and show example Live Site, Abstract, THIC, and Titles Only summaries for APAR (authorized program analysis report) and downloadable-file types of documents.

The programmatic sentence extraction summarizer used for the study implements a well-known sentence extraction model analyzing lexical cohesion factors in the source document text.[4,5] Sentence extraction is driven by the notion of salience—the resulting summary is constructed by identifying and extracting the most salient sentences in the source document. The salience score of the sentence is defined partly from the salience of the vocabulary items in it, and partly from its position in the document structure and the salience of surrounding sentences.

THIC summary. THIC summaries consist of one or more text *snippets*, each illustrating an instance of an occurrence of one or more search terms in a document. (See, for example, Reference 11.) By snippet we mean a text fragment. Thus, the construction of a THIC summary depends upon user-entered search terms. In our study, the search terms were not selected by the participants, but were chosen in advance for each search task by the study team. (See the section "User study method," for details on how the search terms were chosen and the rationale for choosing them in advance). Nonetheless, the THIC summaries were constructed to reflect the preselected search terms. In particular, the preselected search terms appeared in the search query entry field on the search page shown to users, and the THIC summaries depended on these preselected terms just as if a user had entered them.

For the THIC summaries used in our study, we also coded a programmatic solution. In the simplest case of our THIC implementation, each snippet that appears in the document summary consists of a contiguous portion of text from the document in which a particular search term is shown along with surrounding text; that is, a fragment of text before the search term, the search term, and a fragment of text after the term, thus showing the term in the context of its use in the document. Our implementation of THIC began by finding the first occurrence of each search term in the document, and, for each such occurrence, extracting a text snippet showing the term in context. The algorithm extended the edges (the head and tail) of each snippet, if necessary, so that words were not truncated. The search terms in each snippet are highlighted by using boldface as in the example THIC summary for search terms "program database source" that follows:

Figure 3 Examples of Live Site, Abstract, THIC, and Titles Only summaries



Example “Live Site” Summaries

A document of type *APAR*:
[1] PN72528 - OBI MULTIMEDIA MODEM MWAVE PROBLEMS WITH MICROSOFT MAIL (MS MAIL
OBI MULTIMEDIA MODEM MWAVE PROBLEMS WITH MICROSOFT MAIL (MS MAIL

A document of type *Downloadable files*:
[2] JDK Conversion Assistant
A tool to convert from one Java Development Kit (JDK) to another JDK.

Corresponding “Abstract” Summaries *– search terms are boldface if they appear in text chosen for summary*
[1] PN72528 - OBI MULTIMEDIA MODEM MWAVE PROBLEMS WITH MICROSOFT MAIL (MS MAIL
[APAR] Typical symptoms: MS Mail sends the dial string to the COM port and then the system seems to lock up. There's never any activity on the phone line. Pressing Ctl-Alt-Del when MS Mail has the focus sometimes allows things to happen, but the system will eventually lock up ...

Product categories: Software; Application Infrastructure Services; Networking and Communications; Network File Systems and Sharing

[2] JDK Conversion Assistant
[Downloadable files] The JDK **conversion** assistant helps switch from one **Java** Development Kit (JDK) **Java** 1.2.2 vendor implementation to another by modifying the WebSphere Server configuration to use the new JDK.
Product categories: Software; Application Servers; Distributed Application and Web Servers; WebSphere Application Server; JDK/SDK; **Operating system(s):** Multi-Platform; **Software version:** 3.0.2 , 3.5

Corresponding THIC Summaries *– search terms are boldface and shown in context*
[1] PN72528 - OBI MULTIMEDIA MODEM MWAVE PROBLEMS WITH MICROSOFT MAIL (MS MAIL
Abstract OBI MULTIMEDIA MODEM **MWAVE** PROBLEMS WITH MICROSOFT MAIL (MS MAIL Error Description Typical symptoms: MS Mail sends the dial string to the COM port ...
Product categories: Software; Application Infrastructure Services; Networking and Communications; Network File Systems and Sharing

[2] JDK Conversion Assistant
… tool to convert from one **Java** Development Kit (JDK) to another JDK. The JDK **conversion** assistant helps switch from one **Java** Development Kit (JDK) ...
Product categories: Software; Application Servers; Distributed Application & Web Servers; WebSphere Application Server; JDK/SDK; **Operating system(s):** Multi-Platform; **Software version:** 3.0.2 , 3.5

Corresponding "Titles Only" Summaries
[1] PN72528 - OBI MULTIMEDIA MODEM MWAVE PROBLEMS WITH MICROSOFT MAIL (MS MAIL

[2] JDK Conversion Assistant

> . . .great Development Environment that allows you to **program** in a variety of . . . become a member of the **Source** Code Group today! . . .Browse the largest code **database** on the best site. . .

Overlapping snippets (if any) were also merged, thereby producing snippets wherein more than one search term occurs. For example, assume the following document fragment:

> The Sync program on the source system must be running continuously not only to synchronize changes made to the source database by the server but also by other applications.

If the search terms are “program synchronize,” then one THIC snippet of the document, containing the term “program,” might be, “The Sync **program** on the source system must be running continuously. . .,” and a second snippet, containing the term “synchronize," might be, ". . .running continuously not only to **synchronize** changes made to the source database. . .." Because the two snippets contain a common fragment of the document (in this case, the end of the first snippet contains the words "running continuously" as does the beginning of the second snippet), they can be merged to form a single snippet: "The Sync **program** on the source system must be running continuously not only to **synchronize** changes made to the source database. . ."

Our algorithm implemented additional processes to minimize the length of the resultant snippet. (Merging was performed recursively on all resultant snippets and becomes more important when there are more than two search terms.) The resultant snippets were then appended one to another to form the overall document summary. Ellipses appear between snippets, at the front of the first snippet if it did not occur at the beginning of a sentence and at the tail of the last snippet if it ends before the end of a sentence.

Titles Only summary. The Titles Only summary used the titles of the documents from the live site with no additional information.

Other features. The documents in question ranged in size from hundreds to thousands of words (i.e., a difference of one or two orders of magnitude overall). As discussed above, handcrafted Live Site summaries were occasionally less than 100 characters in length. (See Figure 3 for examples.) Other Live Site summaries were as long as 500+ characters. The lengths of the THIC and Abstract summaries (and therefore the amount of compression) were roughly equivalent and ranged approximately from 200 to 500 characters in length. One could thus say that the summary lengths were roughly equivalent except for the Titles Only case.

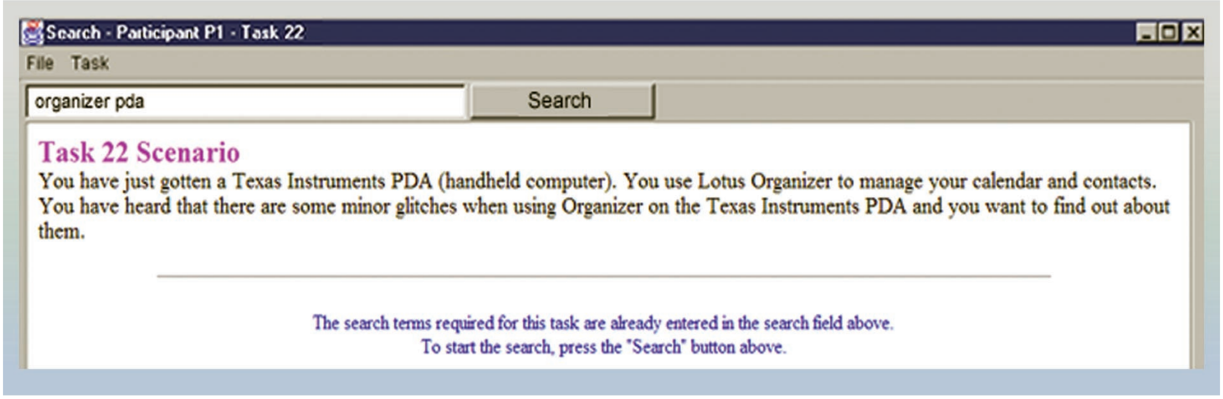
An additional objective was to learn if three other features were helpful to users in deciding which documents were relevant to their needs. These features were use of boldface for search terms, inclusion of document type information, and inclusion of formatted product information in technical support document summaries. The search terms were highlighted in boldface in the THIC summaries and also in Abstract summaries if the terms were present. Product information was taken from a specific section of the documents and included such information as product line, operating system, hardware platform, and version. The precise information included depended on the product. The Abstract and THIC summaries had product information. The Abstract summaries had document type information as well.

Hypothesized advantages of different summary types. As we saw earlier, the Live Site summaries vary substantially. We expected that summaries that were all or part of a human-authored section would fare well if the section chosen matched the information needs of the task. Target documents whose summaries consisted of the first 255 characters were unpredictable; the results would depend on the contents of the characters. Because Abstract summaries were composed of sentences from the document, we thought they might be easier to read than the THIC summaries, which were composed of disconnected snippets of text. Also, if the heuristics for selecting sections to be summarized and the sentence extractor's notions of salience were correct, the Abstract summaries might do a good job of distilling the core content of documents. On the other hand, THIC summaries had the benefit of always containing the search terms, which presumably relate directly to the task at hand. The quality of THIC summaries would depend on the search terms used. We could anticipate no advantage for Titles Only, and thus we hypothesized that Titles Only summaries would fare worse then the other summary types.

User study method

A useful and widely accepted methodological basis for evaluating general summarization systems was established in the 1999 TIPSTER/SUMMAC final report,[12] which presented two approaches to evaluating summarization systems: *intrinsic* (or normative) evaluation that judges the quality of the summary directly, based on analysis of the summary content in terms of some set of norms, and *extrinsic* evaluation that judges the quality of the summarization based on how it affects the completion of some other task. Extrinsic evaluations include question-answering and comprehension tasks, as well as tasks that measure the impact of summarization on determining the relevance of a document to a topic. The two methodologies were also used by Jing, et al.[13] in a study where intrinsic evaluation was accomplished by comparing automatically generated summaries to *ideal* human-crafted summaries, and extrinsic evaluation was based on an information-retrieval task. Other published studies have used only one of the methods for evaluating different summarization systems, in some cases comparing system results to human-crafted summaries[14] or evaluating summaries

Figure 4 Example of the initial screen of the interactive tool used by study participants



Task 22 Scenario

You have just gotten a Texas Instruments PDA (handheld computer). You use Lotus Organizer to manage your calendar and contacts. You have heard that there are some minor glitches when using Organizer on the Texas Instruments PDA and you want to find out about them.

The search terms required for this task are already entered in the search field above.
To start the search, press the "Search" button above.

based on information-retrieval task behavior.[15] Note that both of these methodologies for evaluating summarization systems require human experts, either to craft the ideal summaries or to judge the relevance of search results using summaries under investigation.

In our study, we used an extrinsic evaluation method, conceptually similar to the extrinsic methods presented in Jing, et al.,[13] and Mochizuki and Okumura.[15] Unlike these other studies, our work focuses on a specific information-retrieval situation involving in particular a technical-support-document corpus and information-retrieval tasks focused on technical support scenarios. We also introduce into the study a unique summarization method that combines analysis of the document type and corresponding structure elements along with state-of-the-art sentence extraction. We further extend interpretation of study results by considering a so-called liberal evaluation criterion along with a more traditional correctness criterion, as described later in this paper.

Study design. Different groups of people served as participants in each of the four summary conditions, a between-subjects design. We designed a set of task-based scenarios that involved doing searches on the technical support site. Each scenario described a situation in which a user is attempting to solve some problem or find specific information on the site. An example scenario is shown in Figure 4. We asked study participants to assume the role of the user in these scenarios.

We used eight document types for target documents (the document in each scenario that provides the answer or solution for the task; see the section "Materials" later). Each person received the same 24 tasks, three for each document type. We divided the 24 tasks into three sets so that each document type appeared as the target once in each set. Task order was counterbalanced across participants by varying the set according to a Latin square. There were 10 results per search task and the target appeared about equally often in each position.

Participants. The participants were recruited by a market research agency. All were information technology (IT) professionals who were users of the enterprise technical-support Web site. All had used the site once a month or more in the past six months, and in the past year all had gone to the Web site to seek information about laptop or desktop computers, basic computer software such as e-mail, and complex hardware or software such as servers or management information systems. We chose to exclude home users because IT professionals were the majority of the site's users and accounted for the bulk of business revenue. There were 10 participants in the Abstract group, nine in the Live Site group, nine in the THIC group, and ten in the Titles Only group.

Materials. The corpus consisted of documents in a variety of formats as described above, but for the most part the structure of each is known. Thus, this study can be viewed as contributing to research not only on summarization of technical-support documents, but also more generally on summarization of structured documents.

In order to have search terms (and documents) that were representative of those used by real users, we started with popular search terms used in a recent month on the live technical support site. To create the scenarios, we developed plausible tasks for each of the popular search queries. For example, for the terms "java conversion" the scenario was "You are using the Sun Java Development Kit** but you need to move to the one provided by Microsoft Corporation. You are looking for an easy way to convert your applications to the Microsoft JDK**." Nine tasks had a one-word search term, eight had two search terms, five had three terms, and two tasks had four terms.

We then chose target documents from among the documents returned for actual search queries using those terms. For each task scenario, we wanted to have only one document that contained the answer with respect to the task; that is, one of the documents provided the most helpful or best solution. Of course, using the 10 most relevant query results from the live site could greatly add ambiguity to the results because among these top 10 there may be several documents that can help resolve the user's information requirements. Therefore we manipulated the result lists with the intention that although all the documents in the hitlist might be plausible results, there would clearly be one document that was the most helpful with respect to the task in question. We subsequently realized that in some of our hitlists, more than one document was credibly very helpful; therefore, we viewed results using two criteria regarding which document was correct. (See the discussion on the topic of strict versus liberal criteria in the section "Mean number correct and mean time.") Note that all results in the hitlist shown to study participants were actual results from searches of the live site's document corpus; for some queries, we just substituted among those 10 results some documents that were relevancy-ranked lower than the top 10 by the live site.

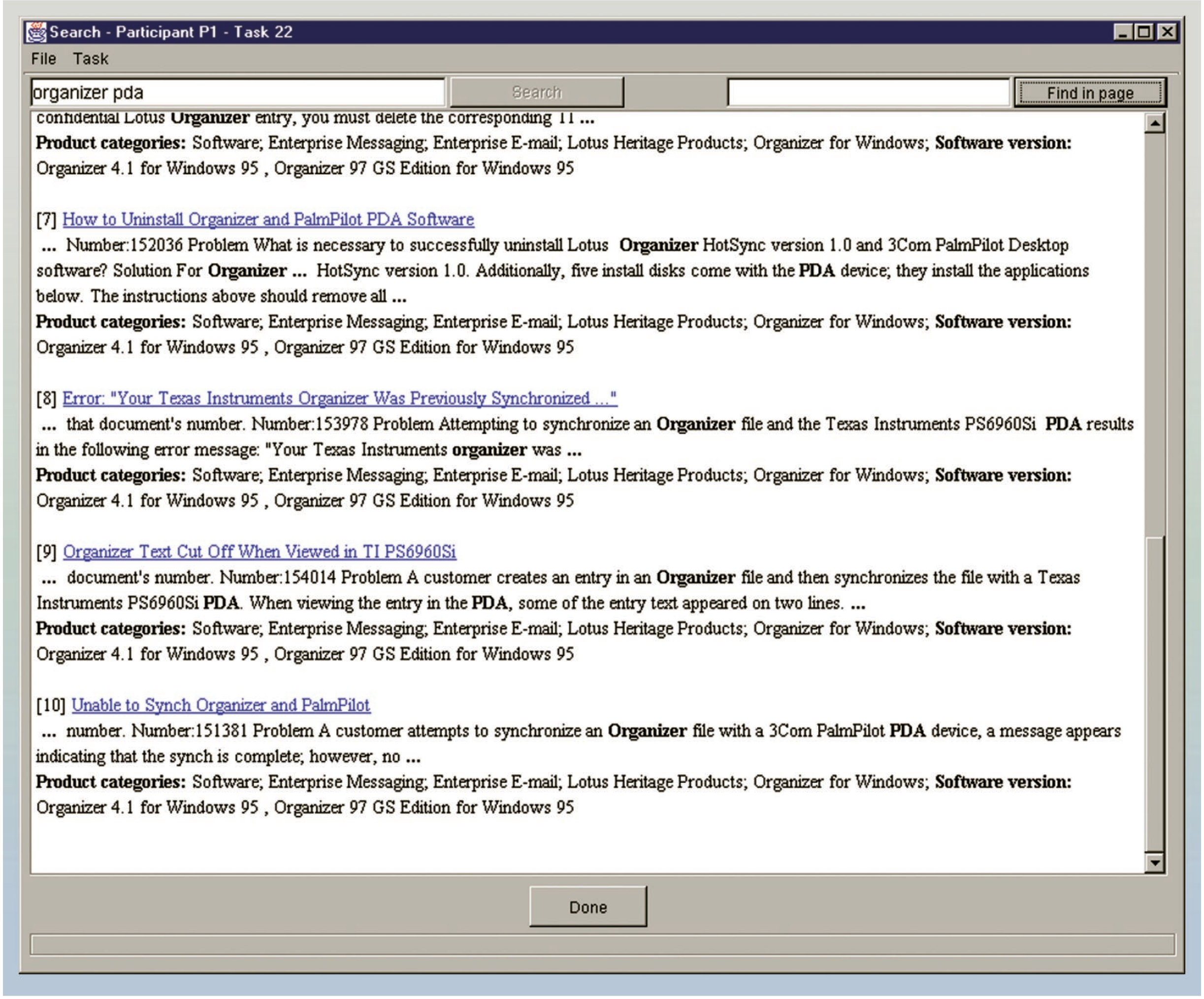
Procedure. Participants interacted with a software tool constructed for the study. This tool presented each search scenario to participants, as in Figure 4. On this initial screen, preselected search terms for the task were seeded by the program in the search-terms entry field. There were a number of reasons for providing this fixed set of system-supplied search terms for each task, rather than having participants enter terms of their own choosing. First, we were not testing participants' query formulation skills, but rather how the content and format of the summaries shown in the result list affected the ability to find relevant and useful documents. Second, all of the search-result pages were created beforehand and then cached in order to ensure that the system response time was equal for all four summary conditions. If the result list had been obtained dynamically from the live site, the extra processing needed to create the other three summary types would have resulted in longer system response times for these conditions. As discussed earlier, we manipulated the cached result hitlists for each search such that a single result would stand out as the best. Also, THIC summaries shown in the cached result pages were based on specific search terms (to allow selection of terms-in-context snippets and the highlighting of the search terms therein). Conditionally, Abstract summaries also included the highlighting of search terms if they appeared in the extracted sentences; thus, we needed to fix the set of terms actually used for the query.

After participants viewed the task scenario on the screen, they pressed the Search button. In response the study application performed the (simulated) search for documents matching the search terms and displayed the search results using one of the four formats as set by the experimenter (Figure 5). Each entry in the result list consisted of a number for its position in the list, the title, which was a hyperlink to the document, and the summary (except in the Titles Only condition). All hyperlinks were live; when a document link was clicked, the actual document was displayed in a second browser window.

We told participants to try to find the best document containing the information needed to resolve the scenario. We asked them to treat the task as a real-world problem and to give it the same diligence as they would a similar problem in their actual work environment. When participants felt they had found the right document, they clicked the Done button (Figure 5) and then entered the number of the chosen document (Figure 6). If participants felt they were unable to find the desired document, they entered a "0" for the document number.

We logged information, including elapsed time, for several events during each task scenario: the start of the search, the point at which the participant indicated (by pressing the Done button the first time) that he or she had decided which was the most useful document given the task scenario, the participant's choice for most useful document, whether that

Figure 5 Example of search results screen of the interactive study tool



confidential Lotus **Organizer** entry, you must delete the corresponding 11 ...
Product categories: Software; Enterprise Messaging; Enterprise E-mail; Lotus Heritage Products; Organizer for Windows; **Software version:** Organizer 4.1 for Windows 95 , Organizer 97 GS Edition for Windows 95

[7] How to Uninstall Organizer and PalmPilot PDA Software
... Number:152036 Problem What is necessary to successfully uninstall Lotus **Organizer** HotSync version 1.0 and 3Com PalmPilot Desktop software? Solution For **Organizer** ... HotSync version 1.0. Additionally, five install disks come with the **PDA** device; they install the applications below. The instructions above should remove all ...
Product categories: Software; Enterprise Messaging; Enterprise E-mail; Lotus Heritage Products; Organizer for Windows; **Software version:** Organizer 4.1 for Windows 95 , Organizer 97 GS Edition for Windows 95

[8] Error: "Your Texas Instruments Organizer Was Previously Synchronized ..."
... that document's number. Number:153978 Problem Attempting to synchronize an **Organizer** file and the Texas Instruments PS6960Si **PDA** results in the following error message: "Your Texas Instruments **organizer** was ...
Product categories: Software; Enterprise Messaging; Enterprise E-mail; Lotus Heritage Products; Organizer for Windows; **Software version:** Organizer 4.1 for Windows 95 , Organizer 97 GS Edition for Windows 95

[9] Organizer Text Cut Off When Viewed in TI PS6960Si
... document's number. Number:154014 Problem A customer creates an entry in an **Organizer** file and then synchronizes the file with a Texas Instruments PS6960Si **PDA**. When viewing the entry in the **PDA**, some of the entry text appeared on two lines. ...
Product categories: Software; Enterprise Messaging; Enterprise E-mail; Lotus Heritage Products; Organizer for Windows; **Software version:** Organizer 4.1 for Windows 95 , Organizer 97 GS Edition for Windows 95

[10] Unable to Synch Organizer and PalmPilot
... number. Number:151381 Problem A customer attempts to synchronize an **Organizer** file with a 3Com PalmPilot **PDA** device, a message appears indicating that the synch is complete; however, no ...
Product categories: Software; Enterprise Messaging; Enterprise E-mail; Lotus Heritage Products; Organizer for Windows; **Software version:** Organizer 4.1 for Windows 95 , Organizer 97 GS Edition for Windows 95

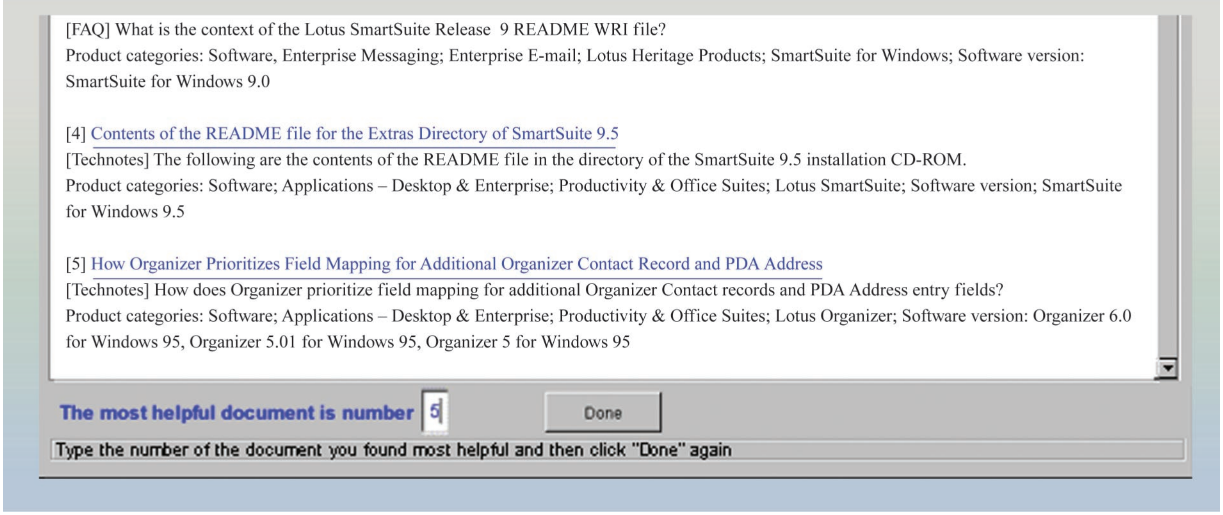
choice agreed with the target, and what document hyperlinks the participant clicked.

User study results

Due to minor problems that occurred with the live site, network connectivity, the study application, and some participants' behavior, not all participants had 24 "valid" tasks—thus, a very small number of user tasks were eliminated from statistical analysis due to problems in collecting valid data for those tasks. Accordingly, in this section we report mean number correct per task. Although there were statistically significant differences due to target document type, we report data averaged across document type. Given the variety of document formats and content within a document type, we could make no meaningful conclusions about document type with only three examples per type. There was no interaction between document type and summary condition. The mean number correct for the four summary conditions and eight document types is shown in Figure 7.

Mean number correct and mean time. The mean number correct and the mean time to complete correct tasks are given in Table 1. The time to complete tasks was measured from the time that participants clicked the Search button to the time they clicked the Done button. We used two scoring criteria: a *strict*

Figure 6 Example of participant identifying the most helpful document for the current task



[FAQ] What is the context of the Lotus SmartSuite Release 9 README WRI file?
Product categories: Software, Enterprise Messaging; Enterprise E-mail; Lotus Heritage Products; SmartSuite for Windows; Software version: SmartSuite for Windows 9.0

[4] Contents of the README file for the Extras Directory of SmartSuite 9.5
[Technotes] The following are the contents of the README file in the directory of the SmartSuite 9.5 installation CD-ROM.
Product categories: Software; Applications – Desktop & Enterprise; Productivity & Office Suites; Lotus SmartSuite; Software version; SmartSuite for Windows 9.5

[5] How Organizer Prioritizes Field Mapping for Additional Organizer Contact Record and PDA Address
[Technotes] How does Organizer prioritize field mapping for additional Organizer Contact records and PDA Address entry fields?
Product categories: Software; Applications – Desktop & Enterprise; Productivity & Office Suites; Lotus Organizer; Software version: Organizer 6.0 for Windows 95, Organizer 5.01 for Windows 95, Organizer 5 for Windows 95

The most helpful document is number 5

Type the number of the document you found most helpful and then click "Done" again

Figure 7 Mean number correct for the four summary conditions and eight target document types

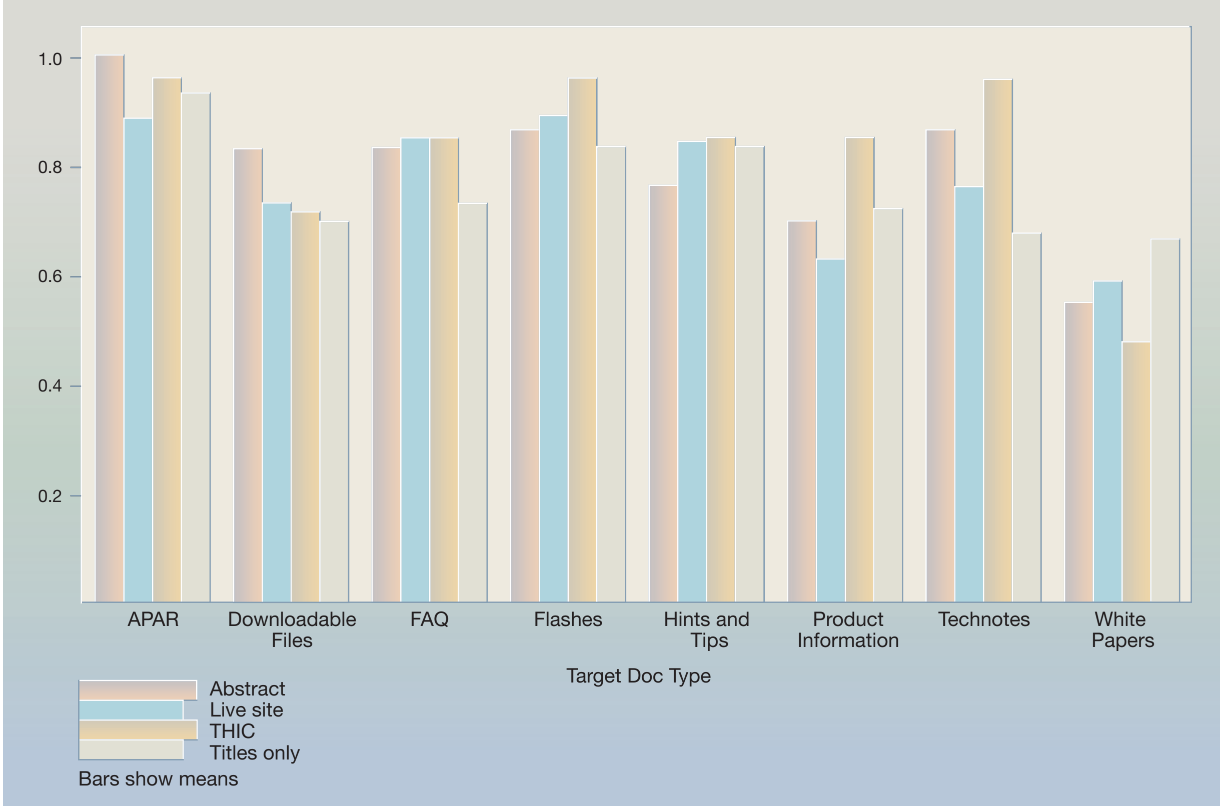


Table 1 Measurements for correctly performed tasks according to strict and liberal criteria

Summary Type	Mean Number Correct (Strict Criterion)	Completion Time in Seconds (Strict Criterion)	Mean Number Correct (Liberal Criterion)	Completion Time in Seconds (Liberal Criterion)
Abstract	.80	92	.87	96
Live Site	.77	84	.84	86
THIC	.83	114	.88	115
Titles Only	.76	87	.82	88

criterion, for which we counted as correct the specific document that we had chosen as best (with the exception of one case in which there was a second document that clearly was equally appropriate). For the other criterion, which we have labeled the *liberal* criterion, we counted as correct any document that nine or more participants chose for the task. Although we had manipulated the hitlist for each search to include a single best hit in our view, there were nonetheless multiple plausibly correct documents in some hitlists—as a result we felt we should create this so-called liberal criterion to perhaps more fairly gauge results. We verified that documents thus rated as correct by the liberal criterion were reasonable choices.

A one-way analysis of variance was done for the four measures in Table 1. With both scoring criteria the same pattern of results emerges. The differences among means for number correct and mean time are not statistically significant. There is a speed-accuracy trade-off for the three types of summaries (Abstract, Live Site, THIC). The means for error rate and speed are inversely ordered.

Because of the variability and skewness of response time data (typically the distribution has a long tail to the high side), we applied the commonly used log transform to the mean time for the strict criterion. The results of analysis of variance on log mean time yielded no significant differences.

Mean number of clicked documents. Another measure of the effectiveness of a summary type is the number of document links that participants clicked on per task. The more informative a summary type is, the fewer documents people will need to read to find the right document. The mean numbers of clicked documents for correct tasks and all tasks are shown in Table 2.

Table 2 Mean number of clicked document links for correctly performed tasks (strict criterion) and for all tasks

Summary Type	Mean Number Clicked for Correct Tasks (Strict Criterion)	Mean Number Clicked for All Tasks
Abstract	1.44	1.77
Live Site	1.72	2.05
THIC	1.78	1.93
Titles Only	1.97	2.31

Although the differences among these means are not statistically significant, the Abstract group had the lowest number of clicked documents and the Titles Only group had the highest. Three people in the Abstract group had a mean number of less than one click for correct tasks. One person in the Titles Only group had a mean less than one; no one in the other two groups had a mean less than one. The mean number of documents clicked for correct tasks and the mean number correct, as well as mean time for the three participants in the Abstract group, are given in Table 3.

The three people in the Abstract group were often able to select the right document based only on the information provided in the hitlist summaries and did so with the same accuracy as the average. Two had considerably lower times than average and one was statistically indistinguishable from the mean for the Abstract group.

In contrast, the participant in the Titles Only group with mean number of clicked documents less than one had the highest error rate of the Titles Only group and the lowest time for correct tasks. Unlike

Table 3 Data for participants in the Abstract group with mean number of clicked document links less than 1 according to the strict criterion

Mean Number of Clicked Document Links for Correct Tasks	Mean Number of Correct Tasks	Mean Completion Time for Correct Tasks (In Seconds)
.39	.75	66
.70	.83	61
.10	.83	104

Table 4 Multiple regression predicting number correct and partial correlations of times and number of clicked documents for each summary type and overall

Summary Type	R	Completion Time for Correct Tasks	Time for All Tasks	Number of Clicked Document Links
Abstract	.86	.71	−.53	−.11
Live Site	.30	.29	−.30	.15
THIC	.65	.61	−.57	−.14
Titles Only	.68	.47	−.38	.11
Overall	.63	.48	−.36	−.05

Table 5 Correlations between measurements for Abstract group

	Number Right	Completion Time for Correct Tasks (In Seconds)	Time for All Tasks (In Seconds)	Number of Clicked Document Links
Number Right	1.00	.80	.69	.24
Completion Time for Correct Tasks	.80	1.00	.97	.42
Time for All Tasks	.69	.97	1.00	.44
Number of Clicked Document Links	.24	.42	.44	1.00

the three participants in the Abstract group with mean number of clicked documents less than one, this person sacrificed accuracy in clicking on few documents.

Table 6 Correlation between measurements for THIC group

	Number Right	Completion Time for Correct Tasks	Time for All Tasks	Number of Clicked Document Links
Number Right	1.00	.32	.15	−.23
Completion Time for Correct Tasks	.32	1.00	.96	−.13
Time for All Tasks	.15	.96	1.00	−.08
Number of Clicked Documents	−.23	−.13	−.08	1.00

Predictors of number correct. In order to gain some insight into the factors related to the number correct, we did a multiple linear regression using time for correct tasks, time for all tasks, and number of clicked documents for correct tasks. Table 4 shows the Pearson multiple R, which represents how well all factors combined predict the number right, and the partial correlations of each factor with number right, which are in turn the correlations of each factor with number right when the effects of all other factors have been removed.

The highest R is for the Abstract group, which means that the combination of time for correct tasks, time for all, and number of clicked documents predicts the number right better for this group than for the others. The proportion of variance accounted for in number right is given by R squared. For the Abstract group this number is 74, which is fairly high for experiments such as this one. The partial correlation of number right with time for correct tasks for this group is the largest of the partial correlations. This means that the relationship between the three predictors and number right is due primarily to the "completion time for correct tasks" parameter. In other words, people in the Abstract group who took longer on correct tasks tended to do better. The same general pattern of results was observed for the THIC and Titles Only groups. To aid in understanding the multiple regression results, the correlations between measures for the Abstract group is shown in Table 5 and for the THIC group in Table 6. The correlation between mean number correct and mean time for correct tasks is statistically significant for the Abstract group but not for the THIC group. As can be seen from Table 5, the number of clicked documents is

Table 7 Mean ratings for questions about helpfulness of information and confidence in choice (1 is most helpful/confident). Means with * are significantly different at the p < .06 level

Summary Type	Helpfulness of Information for Last Task	Confidence in Choice for Last Task	Helpfulness of Information Overall	Confidence in Choice Overall
Abstract	1.98	1.83*	2.10	2.20
Live Site	2.41	2.44	2.44	2.44
THIC	2.41	2.15	2.63	2.33
Titles only	2.77	2.87*	2.90	2.40

moderately correlated with time for correct tasks but has a low correlation with number right.

Helpfulness of information and confidence in choice. After every four tasks, participants were asked the following questions:

- For the task just completed, how helpful was the information in the search results list?
- For the task just completed, how confident are you that you found the right document?

For the first question we used a seven-point scale, ranging from "very helpful" (a rating of 1) to "not at all helpful." For the second question a seven-point scale, ranging from "very confident" to "very unconfident," was used. We asked the same questions at the end of the session for the tasks overall. Table 7 gives the mean ratings for these questions for the four summary types.

For the task just completed, the confidence rating for the Abstract group is significantly better than the rating for the Titles Only group. Although the other differences are not statistically significant, the trend is clear: Titles Only has the worst rating in three out of four cases, whereas Abstract always has the best rating among all summary types.

Boldface search terms, product information, and document type. After they had finished all tasks, we asked participants about the helpfulness of including the product information that appeared at the end of our Abstracts and THIC summaries (see Figure 3), showing the search terms in boldface in the summaries, and including the document type information that appeared in our Abstract summaries. Table 8 shows the means for the groups to which these questions applied. The same seven-point scale was used, with 1 being the most favorable rating. T-tests were used to evaluate the statistical significance of the differences between means for boldface search terms and product information. There were no significant differences between means. All features received moderately favorable ratings.

Table 8 Mean ratings for product information, boldface search terms, and document type information

Summary	Product	Boldface	Document Type
Abstract	2.90	3.50	2.11
Live Site	N/A	N/A	N/A
THIC	2.89	2.33	N/A
Titles Only	N/A	N/A	N/A

More insight into the value of these features came from comments participants made. A number of people said that operating system information was the most important product information. Several also said it was useful to separate hardware, software, and driver problems. A number said that drivers were the most important document type.

Other comments. Participants made a variety of comments when asked how search could be improved and what they liked about search on other sites. The comments have to do with both document content and search features. Many said that documents, summaries, and titles should be in clearer language rather than the occasional technical jargon that appeared in our study documents. Titles should be more descriptive, and it should be easier to see at a glance what a document contains.

Many participants said they liked Google and wanted features provided by that site. They mentioned specifically that they wanted to be able to search within results and to refine searches. Some said that search should be more intelligent. For example, they asked that the search engine show results that might match not only the chosen search terms but also other terms with the same meaning. Another said that if one is in a particular product or category and then clicks “drivers,” only drivers for that specific product should be shown, not the results of a general driver search. Some stated that other companies’ sites track which products they own and use that information in search applications.

Observed strategies for examining results. We observed that a number of participants moved the cursor from title to title, sometimes stopping to read the summary. Others moved the cursor from title to title but also pointed at boldface search terms. We infer from this behavior that participants scanned the titles and looked for boldface search terms before deciding whether to read the summary. It appears that summaries were read by users when the associated titles or highlighted terms seemed promising; that is, a potentially useful document was indicated. A subsequent (unpublished) search study in which participants were encouraged to talk about what they were doing confirmed these behaviors.

Discussion and implications for technical support search

The success rate in this study was higher than that found for searches on the actual enterprise Web site due to our need to create scenarios that could be described in a few sentences, whose goals were clear, and where a solution definitely existed among the set of documents shown in the hitlist. With lower success rates, it is possible that the differences found in this study might be magnified.

When users search a technical support Web site for problem-solving purposes, they know the symptoms of the problem and focus on finding a document that matches those symptoms. The expert authors in the document-author study composed summaries by using sentences from the specific section of the document that described the problem, rather than from the entire document. When the obvious section was inadequate, they used their human intelligence to obtain additional sentences from another section of the same document, or in some cases a linked document, but selected such additional information carefully so as to support the task-oriented nature of the summary. The Abstract summaries in our experiment followed this general task-orientation heuristic and fared better overall than the other summarization approaches. The first implication of our results for technical-document summarization is that programmatically generated summarization should follow such task-oriented strategies.

Looking beyond the domain of technical support, it is clear that people may search an enterprise Web site for many different reasons involving a variety of tasks and information needs. As Kan and Klavans[16] point out, different parts of the same document may serve different information needs. Both explicit and implicit techniques that aid in understanding the user’s task can be used to help select the right section of the document for the summary. Implicit methods, such as tracking the user’s navigation and capturing click stream data, can be used to infer the user’s task. Explicit methods, such as asking a user to select a task from a small set of tasks, for example, “general product information,” “product specifications and prices,” or “troubleshoot a problem,” can also be used to help decide the section of a document to use for a task-oriented summary.

In the user study, the Titles Only and Live Site conditions resulted in shorter times and lower accuracy, but none of the differences were statistically significant. There appears to be a trade-off between speed and accuracy in that the Titles Only and Live Site groups had the worst accuracy and THIC the best, but for time to complete correct tasks the order was reversed. In general, the summary for the Live Site was shorter than that for the Abstract or THIC. Titles Only had no summary and so had the briefest information in the result list. This probably explains the finding that accuracy was lowest for Titles Only and the Live Site, but speed was fastest. The trade-off between speed and accuracy was also evident within groups, in that participants who had longer times for correct tasks tended to get more correct answers. However the correlation was only statistically significant for the Abstract group. Although we must be careful not to mistake correlation for causality, it may suggest that time spent reading the Abstract summaries pays off in increased goal attainment.

With regard to the trade-off between speed and accuracy for search facility design, one might start with the perhaps specious assumption that the best possible summary would be very short and completely informative, but come to understand that this cannot realistically be achieved. However, there are more practicable implications that we can derive from our study. For example, our results indicate that an effort to provide terse summaries so that users may scan them more quickly is not necessarily an appropriate solution: users may be able to view short summaries more rapidly but will achieve lower accuracy in selecting correct result documents. This in turn will cause them to iterate on scanning the hitlist and viewing documents; the obvious end result is that overall time on task will not be shorter—users will spend less time reading summaries but lose that time savings in viewing multiple documents. On the other hand, more comprehensive and, when possible, task-oriented summaries require a bit more time to read initially as the user scans a hitlist but will lead to overall time savings because multiple documents will not have to be viewed. To us, this latter solution is more appealing because it leads to greater user satisfaction.

Participants in the programmatically generated Abstract group were more confident that they had found the right document than those in the Titles Only group. The Abstract group had the best ratings on all four judgments of helpfulness and confidence, whereas the Titles Only group had the worst scores on three out of four (although again the differences were not statistically significant). We feel that these favorable ratings for the Abstract summaries—particularly the sense of confidence on the user's part—should translate into greater user satisfaction. User preferences of this sort may, in fact, be just as important as task performance in adoption of a site by customers.

We also looked at the average number of documents clicked for correct tasks. Each hitlist item contains a document title which serves as a hyperlink to the actual document. In a problem-solving scenario, users click on these document links when it is either clear or probable, based on the hitlist summary, that the document provides the information necessary to solve the problem. Note that in our experiment and even in a real-world setting, users may not click on a summary's document link if the required information for the task is found in the summary itself. In our experimental setting, users may elect not to click on a document link in the clear case, meaning the participant is confident that the most useful document has been discovered based upon its summary alone. Thus fewer document-link clicks indicate more confidence in making a correct document choice based only upon the information provided by the summaries.

The results of the click-through analysis favored the Abstract condition in that three subjects in this group had an average of less than one document-hyperlink click. Their accuracy was about the same as the mean. In other words, these three people in the Abstract group were often able to choose the right document by using only the information in the summary, without viewing the document itself and without sacrificing accuracy. The multiple regression analysis confirmed that the number of clicked documents was not correlated with accuracy for any group. These three participants also had task completion times that were faster or the same as the mean for the group.

The number of clicked documents had a moderate positive correlation with time for correct tasks, suggesting that people who clicked more documents had longer task times. The Titles Only group had the highest number of clicked documents, but the differences between means were not statistically significant. In real life people would probably spend more time than in this study looking at each document clicked, so the time savings for looking at fewer documents might be greater. This analysis further supports the ideas outlined earlier regarding implications of the speed-accuracy trade-off.

Given the overall pattern of results, we can reasonably conclude that Titles Only is inferior to the programmatic Abstract. It is interesting and perhaps surprising that the participants in Titles Only group did as well as they did. We observed in the other groups that many people seemed to scan the titles before deciding which summaries to read. That is probably an efficient strategy because it allows the user to eliminate many candidates without reading the summaries. The clear implication for search (on technical support sites and in general) is that documents must have titles that are highly descriptive of the document content.

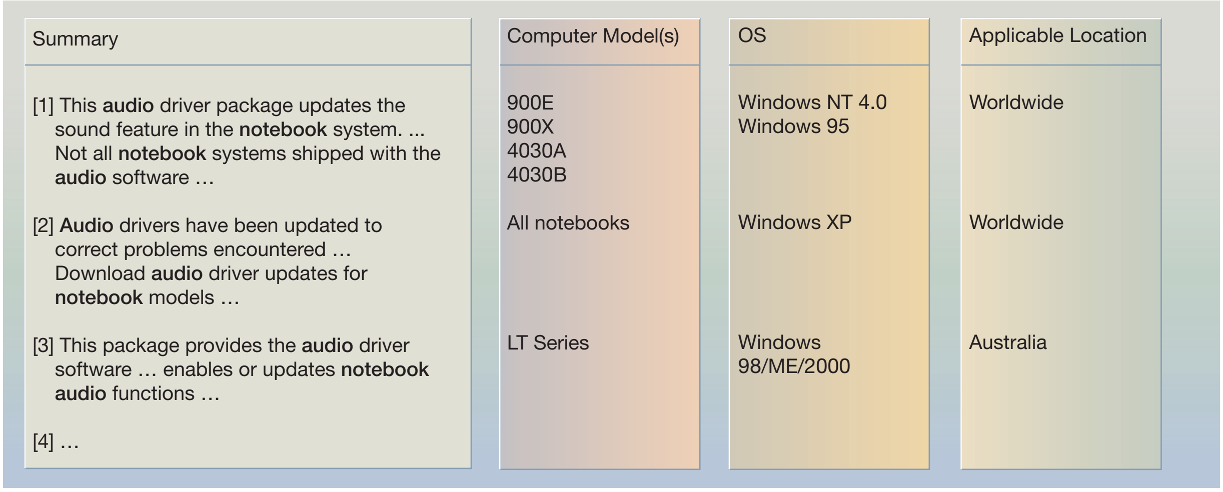
This work demonstrates that using information about the subcomponent structure of documents to guide selective extraction can result in more useful document summaries for search users. Using sentences from specific sections that are known beforehand to be the most meaningful and useful portions of individual documents, especially with respect to users' task-based information needs, appears to be a favorable approach for search result summaries. We stated earlier that although human-generated summaries would result in the most semantically meaningful summaries, this approach is untenable from a cost perspective for legacy corpora containing hundreds of thousands or millions of documents. If, on the other hand, an enterprise corpus contains documents that are formatted according to a set of discernable rules, styles, or templates, as was the corpus we studied, a useful and cost-effective human intervention is to derive a set of heuristics for programmatically extracting the most relevant information from those structured documents. In the case of technical support documents in the corpus of a particular enterprise, the content structure of documents (for example, section names, which are semantically meaningful to people) should be a known entity, and enterprise-specific, document-type-specific programmatic summarization algorithms should take full advantage of this semantically useful information. Thus summaries that more accurately reflect the task-oriented content of support documents are achievable. Such summaries make users feel more confident of finding the desired results quickly, without the frustrating need for and wasted time involved in numerous restarts (returning multiple times to the hitlist and viewing multiple documents).

Given that the THIC approach seems to be the method of choice for many general Web search engines, it is perhaps surprising that it fared no better than the other methods. THIC summaries presumably contain relevant portions of the document because the text is dynamically selected based on, and centered on, the search terms for a task. Creating summaries of documents for a technical support Web site, however, is different from the task of creating summaries for any and all documents on the Web. Although the structure of these technical documents was varied, it was known beforehand, and the Abstract method was able to use heuristics that took advantage of this structure. For example, the heuristics for the Abstract summaries often selected the document sections that described the problem or question asked. In other words, they were based upon the task for which the document was designed. In contrast, general Web search engines must handle unpredictable structure, and a single document may contain many topics of no interest to the person searching. It appears that for technical support documents, the benefits of basing the summary on the search terms (as in THIC summaries) are balanced by the benefits of basing the summary on the task for which the document was intended. It is also possible that the Abstract summaries were easier to read than THIC summaries because the former consisted of complete sentences while the latter consisted of disconnected snippets of text. In future work, it would be interesting to compose a summary using the THIC method, but with text only from the sections specified in the structure-based Abstract heuristics.

The Abstract approach we have outlined here can, at least currently, only practicably apply to specific corpora containing documents of a limited set of styles and subcomponent structure. Other popular and more general summarization techniques such as THIC are completely ignorant of document structure or the importance or centrality of particular document sections to overall document semantics, but our Abstract method involves prior analysis by a human being of the structure of the documents and the content located in specifically named subsections of documents. Overall, our approach is thus based on human analysis of the document structure, content location within documents, and the usefulness of specifically located content for task resolution. To achieve this result, summarization has to not only focus on the textual content of a document but also capitalize on knowledge beforehand of a document's content structure. For Web search in general it is impossible to capitalize on such a priori knowledge. Nonetheless, using linguistics-based and knowledge-based techniques, summarizers can attempt to capitalize on the linguistic and semantic information that can be extracted from documents, such as meaningful section titles, as well as metalinguistic information, such as HTML or XML (eXtensible Markup Language) tag names and tag content (e.g., the words used in HTML heading tags), or other structural clues, in an attempt to replicate this more intelligent summarization approach. Others, in fact, have attempted summarization techniques based much more generally upon discourse-based document structure. For example Ono, Sumita, and Miike[17] and Marcu[18] have attempted to extract a document's rhetorical structure for summarization purposes—this approach seems to work quite well for short documents but perhaps not as well for large texts.[19]

Another perspective regarding implications of this study is the following. Based on our results, programmatically generated text-only summaries all result in somewhat differentiated but similar utility for users. Kan and Klavens[16] also found no significant differences due to summary type in a comparison of multiple document summaries versus single document summaries (a different issue from the subject of the current study but with the same conceptual result). Perhaps both to provide the most salient and relevant information contained in documents and to distinguish documents from one another in terms of their informational relevance to a search query, search facilities ought to provide information in other than plain-text form. Clustering of results into informational categories is one possibility, and, indeed,

Figure 8 Example of result list incorporating textual summaries and additional information for distinguishing hitlist documents from one another



Summary	Computer Model(s)	OS	Applicable Location
[1] This **audio** driver package updates the sound feature in the **notebook** system. ... Not all **notebook** systems shipped with the **audio** software …	900E 900X 4030A 4030B	Windows NT 4.0 Windows 95	Worldwide
[2] **Audio** drivers have been updated to correct problems encountered … Download **audio** driver updates for **notebook** models …	All notebooks	Windows XP	Worldwide
[3] This package provides the **audio** driver software … enables or updates **notebook** **audio** functions …	LT Series	Windows 98/ME/2000	Australia
[4] …			

Dumais et al.[20] have shown the value of grouping search results in categories. In computer support applications, our participants' comments suggest that grouping search results by operating system or hardware platform should make the search process more accurate and faster because users need only look at the applicable category. The benefit of categorizing search results should be greatest when there are a large number of results. Here, clustering is based on distinguishing semantic features of the documents' content. Going a step further, clustering might be performed according to features related to the user's search terms; that is, items in the hitlist would be clustered into salient categories where the categories themselves are dynamically selected based on features of the user's search terms.

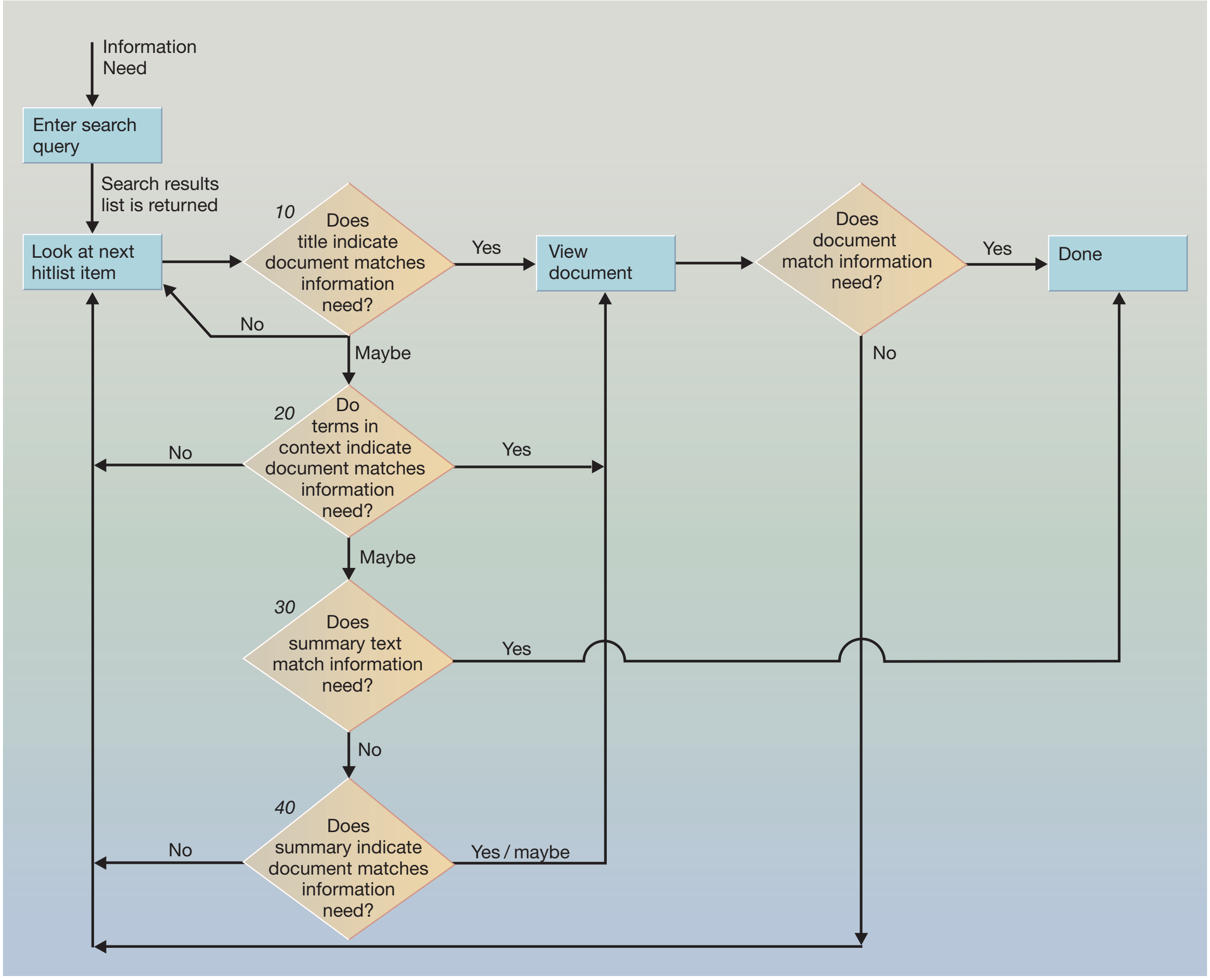
Perhaps additional types of information visualization—instead of or in addition to text-only summaries—are required to provide for both speed and accuracy of assessing search results. We have also been experimenting with visualization techniques such as tables and other types of structured formatting that display, in readily recognizable forms, the features that distinguish hitlist documents from one another in addition to textual summaries for each document. The information displayed in such visualizations can be extracted based upon relevance to the user's entered search terms, upon features in documents that possess lexical or semantic relationships to the search terms, or upon meaningful feature terms in documents that are in close proximity to search terms. Thus, like THIC, such hitlist information may be dynamically generated based on the user's query. For example, Figure 8 shows a table that incorporates THIC summaries for each document in the hitlist, along with distinguishing information based on search terms ("notebook audio" in this case) and related terms.

As differences among competitive products decrease and the quality of support and service become increasingly important, we expect to see more efforts aimed at improving customer goal attainment on technical support Web sites that will also serve to further the state of the art and practice in search, summarization, and document authoring tools. We must be cautious in generalizing our findings beyond the domain of technical support and the particular parameters and conditions studied in this work, particularly since most differences among means were not statistically significant. However, we see promise in approaches to summarization that take advantage of structure of documents and the tasks for which the documents were intended.

A process model for search

On the basis of the quantitative data and observed behavior in the user study, we conclude by proposing a process model to describe the behavior of participants in our user study (Figure 9). This model

Figure 9 An approximate model of user behavior when searching and evaluating results with respect to a specific information need



Information Need
Enter search query
Search results list is returned
Look at next hitlist item
10
Does title indicate document matches information need?
Yes
View document
Does document match information need?
Yes
Done
No
Maybe
No
20
Do terms in context indicate document matches information need?
No
Yes
Maybe
30
Does summary text match information need?
Yes
No
40
Does summary indicate document matches information need?
No
Yes / maybe

facilitates further insight into additional implications of this work.

Users start with an information need; in the case of our study this was described in the scenario and was already translated into a query—of course, in a real world situation, the query term selection would be done by the user. On the search-result page, users often read (only) the titles of hitlist items looking for a potential match to their information need. For example, in cases in which they have a specific problem to be solved, they may look for symptoms of the problem in the title. This may explain why the Titles Only approach was not significantly worse than the other summary types. Users may also initially look for boldface search terms in the summary. If it appears that the item may satisfy the information need, users then read (or scan) the summary; if not, they go on to the next hitlist item. If the summary appears to indicate a document that contains the appropriate information required to complete the task, users then look at the corresponding document to determine if it indeed contains the necessary and sufficient information for their information needs. In fact, the summary itself may contain the necessary and sufficient information to satisfy the information need; for example, the user may be looking for an e-mail address or phone number that appears in the summary—and in this case viewing the document itself is unnecessary.

Of course, there are other variations on this basic process. For example, some participants read several titles, assessing the relative likelihood that the documents had the desired information, before deciding which summaries to read; some users read the title and then scanned the summary, skipping the "look for boldface terms only" step, and so on. There are also variants of this process that are elided from the diagram for clarity purposes; for example, a user may go directly from box 10 to box 30 or 40 in Figure 9.

An important aspect of the model is that participants typically read only those summaries that look promising, based on titles and, perhaps, the presence and textual context of boldface search terms. Therefore, it may be that the differences among the summary types tested in this study may be second-order effects. This might explain the finding that although the Abstract group had the most favorable values on a number of measures, these measures usually did not reach statistical significance. Although there are many differences between this experiment and real-life situations, we believe that the model captures realistic behavior regarding the reading of summaries in search-result hitlists.

Acknowledgments

We thank the anonymous reviewers who provided many useful suggestions toward the improvement of this paper.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Google, Inc., Overture Services, Inc., Sun Microsystems, Inc., or Microsoft Corporation.

Cited references

1. K. Ehrlich and D. Cash, "Turning Information into Knowledge: Information Finding as a Collaborative Activity," *Proceedings of the First Annual Conference on the Theory and Practice of Digital Libraries*, Texas A&M University, College Station, TX (1994), pp. 1–8.
2. Accenture, http://www.accenture.com/.
3. Apple Computer, Inc., http://kbase.info.apple.com/.
4. B. Boguraev and M. Neff, "Lexical Cohesion, Discourse Segmentation and Document Summarization," *Proceedings of the RIAO 2000 Conference on Content-Based Multimedia Information Access*, Paris, France (2000), pp. 962–979.
5. T. Hand and B. Sundheim, "TIPSTER/SUMMAC Summarization Analysis. Tipster Phase III 18-Month Meeting," *Working Papers from the SUMMAC Conference*, NIST, Fairfax, VA (1998).
6. H. Jing and K. McKeown, "The Decomposition of Human-Written Summary Sentences," *Proceedings of the 22nd Conference on Research and Development in Information Retrieval (SIGIR'99)*, Berkeley, CA, ACM, New York (1999), pp. 129–136.
7. E. Amitay and C. Paris, "Automatically Summarizing Web Sites: Is There a Way Around It?," *Proceedings of the ACM 9th International Conference on Information and Knowledge Management (CIKM 2000)*, Washington, D.C., ACM, New York (2000), pp. 173–179.
8. D. R. Radev, "Text Summarization," http://www.summarization.com/.
9. Summarization Bibliography: http://www.csi.uottawa.ca/tanka/ArtDB/bibliography.html.
10. H. Jing, "Summarization Resources," http://www.cs.columbia.edu/~hjing/summarization.html.
11. S. Lawrence and C. L. Giles, "Context and Page Analysis for Improved Web Search," *IEEE Internet Computing* **2,** No. 4, 38–46 (1988).
12. I. Mani, D. House, G. Klein, L. Hirschman, L. Obrst, T. Firmin, M. Chrzanowski, and B. Sundheim, "The TIPSTER SUMMAC Text Summarization Evaluation: Final report," *DARPA Technical Report* (1999), http://citeseer.nj.nec.com/mani99tipster.html.
13. H. Jing, R. Barzilay, K. McKeown, and M. Elhadad, "Summarization Evaluation Methods: Experiments and Analysis," *Working Notes of the AAAI-98 Spring Symposium on Intelligent Text Summarization*, Stanford, CA, AAAI, Menlo Park, CA (1998), pp. 60–68, http://citeseer.nj.nec.com/jing98summarization.html.
14. J. Klavans, K. McKeown, M. Kan, and S. Lee, "Resources for the Evaluation of Summarization Techniques," *Proceedings of the 1st International Conference on Language Resources and Evaluation*, Grenada, Spain (1998), http://citeseer.nj.nec.com/5308.html.
15. H. Mochizuki and M. Okumura, "A Comparison of Summarization Methods Based on Task-Based Evaluation," *Second International Conference on Language Resources and Evaluation (LREC-2000)*, Athens, Greece (2000), pp. 633–639, http://citeseer.nj.nec.com/406965.html.
16. M.-Y. Kan, and J. L. Klavans, "Using Librarian Techniques in Automatic Text Summarization for Information Retrieval," *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL 2002)*, Portland, OR, ACM, New York (2002), pp. 36–45, http://www.cs.columbia.edu/~pablo/community/nlp/jcd102.pdf.
17. K. Ono, K. Summita, and S. Miike, "Abstract Generation Based on Rhetorical Structure," *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, Kyoto, Japan, ACL, East Stroudsburg, PA (1994), pp. 344–348.
18. D. Marcu, D., "The Rhetorical Parsing of Natural Language Texts," *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, Spain, ACL, East Stroudsburg, PA (1997), pp. 96–103.
19. L. Carlson, M. E. Okurowski, J. M. Conroy, D. Marcu, W. Wong, D. P. O'Leary, and A. Taylor, "An Empirical Study of the Relation between Abstracts, Extracts, and the Discourse Structure of Texts," *Proceedings of the 1st Document Understanding Conference (DUC 2001)*, New Orleans, LA, NIST, Gaithersburg, MD (2001).

20. S. Dumais, E. Cutrell, and H. Chen, "Optimizing Search by Showing Results in Context," *Proceedings of the CHI 2001 Conference on Human Factors in Computing Systems*, Seattle, WA, ACM, New York (2001), pp. 277–284.

Accepted for publication February 9, 2004.

Catherine G. Wolf *Research Division, IBM Thomas J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598 USA (cwolf@us.ibm.com).* Dr. Wolf received a Ph.D. degree from Brown University in psychology. She has investigated a range of issues in human-computer interaction since coming to the Watson Research Center 18 years ago. In addition to her work on search, she has worked on handwriting and gestural interfaces, conversational interfaces, and collaboration. Dr. Wolf has published widely in the field of human-computer interaction and holds a number of patents.

Sherman R. Alpert *Research Division, IBM Thomas J. Watson Research Center, PO Box 218, Yorktown Heights, NY 10598 (salpert@us.ibm.com).* Mr. Alpert has been with the Watson Research Center since 1987. He received a B.S. degree in computer science from the State University of New York at Stony Brook and an M.A. degree in computing education from Columbia University Teachers College, where he has pursued additional studies toward a doctorate. He has been involved in research and software development in a variety of domains including educational technology, human-computer interaction, multimedia, and object-oriented programming and design, and has published widely in these fields. He serves on the editorial and review boards of several journals and on a number of conference and program committees.

John G. Vergo *Research Division, IBM Thomas J. Watson Research Center, PO Box 218, Yorktown Heights, NY 10598 USA (jvergo@us.ibm.com).* Mr. Vergo is a member of the Technical Strategy team at the Watson Research Center. His research interests include human-computer interaction, User-Centered Design methods, multimodal user interfaces, e-commerce user experiences, speech recognition, natural language understanding, scientific visualization, 3D graphics, and software development methods. He has a B.S. degree in mathematics and psychology from the University at Albany, State University of New York and an M.S. degree in computer science from Polytechnic University, Brooklyn, NY.

Lev Kozakov *Research Division, IBM Thomas J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598 USA (kozakov@us.ibm.com).* Dr. Kozakov is a research staff member at the Watson Research Center. He received a Ph.D. degree in applied mathematics and computer sciences from M.V. Lomonosov Moscow University in 1983. He has worked in many areas, including dynamic systems, applied statistics, information management systems, man-machine interface, and object-oriented design and programming. His current research interests include information management frameworks and natural language processing technologies. He has a number of publications in various fields of computer science and applied mathematics and holds several patents.

Yurdaer Doganata *Research Division, IBM Thomas J. Watson Research Center, PO Box 704, Yorktown Heights, NY 10598 USA (yurdaer@us.ibm.com).* Dr. Doganata is the manager of the Information Management Solutions Group at the Watson Research Center. He received B.S. and M.S. degrees from the Middle East Technical University, Ankara, Turkey and a Ph.D. degree from the California Institute of Technology, Pasadena, California, all in electrical engineering. He joined the Watson Research Center as a research staff member in 1989 and has worked on and managed projects in many diverse areas, including high-speed switching systems, multimedia servers, intelligent transportation systems, multimedia collaborative applications, e-services, and information search and retrieval systems for technical support. His current work involves designing and prototyping innovative solutions, applications, tools, and utilities in the area of unstructured information management.