Playing Detective with Full Text Searching Software

Darrell R. Raymond
Heather J. Fawcett

Centre for the New Oxford English Dictionary
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

ABSTRACT

Searching large text databases often resembles detective work. We explored this notion with an experiment in which subjects used powerful full text searching software to solve problems about the Arthur Conan Doyle story *The Hound of the Baskervilles.* The experiment was conducted in two parts: in the first part subjects attempted to teach themselves about the software using only the documentation; in the second part, subjects used the software to answer questions such as *What brand of cigarette does Watson smoke?* The experiment provided a great deal of feedback about the usability of the software and the documentation. Among the results that have wider implications are the need for better display of context, and a need for careful documentation of the characteristics of full text searching.

1. INTRODUCTION.

> *"I have in my pocket a manuscript," said Dr. James Mortimer.*
>
> *"I observed it as you entered the room," said Holmes.*
>
> *"It is an old manuscript."*
>
> *"Early eighteenth century, unless it is a forgery."*
>
> *"How can you say that, sir?"*
>
> *"You have presented an inch or two of it to my examination all the time you have been talking. It would be a poor expert who could not give the date of a document within a decade or so."*

Sherlock Holmes may be able to tell everything about a document by observing only an inch or so, but most people need to see more of the document than that. We confirmed this hypothesis during an experiment involving PAT, a full text searching system constructed at the University of Waterloo for use with the online Oxford English Dictionary.[1] Full text searching is becoming a popular means of accessing online text, partly because the necessary processing power is now widely available and partly because semantic indexing of text is still an expensive process. However, full text systems place much of the onus of searching on the user, who must supply lexical strings that are likely to be used in the relevant text. How this process operates in the context of searching online documentation has not been well studied.

An important goal in the present experiment was to evaluate the usability of PAT and its user manual.[2] Although PAT has been successfully and extensively used over the last two years to search a variety of texts, all evaluations to date have been extremely informal. The interface to PAT at the time of testing was an interactive command-line system with a simple concordance display of the text. Figure 1 shows a screen dump of a typical PAT session. The user manual is task-oriented, making full use of examples wherever possible. Figure 2 shows the layout of the user manual.

Our second reason for conducting an experiment was to discover more about the process of full text searching. In particular, we were somewhat baffled that other studies [3,4] had indicated grave deficiencies in full text searching, while our experiences seemed to indicate that full text searching was quite useful. We were particularly interested in situations in which users seek answers to specific questions, such as would be common in searching online documentation.

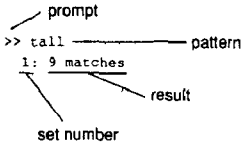
We planned to have our subjects search a single large text. The experiment was conducted in two separate sessions: a training session and a searching session. In the training session, subjects were exposed to the documentation and required to solve some simple problems. In the

```
% pat /usr/newoed/holmes/houn
Pat 3.3  Text searching system
Copyright 1989 University of Waterloo

>> "my dear watson"
  1: 10 matches

>> beard
  2: 18 matches

>> pr sample.5
   249057, ..eard!" <p> "A beard?" <p> "It is not the baronet -- it is -- why..
    67209, ..f bushy black beard and a pair of piercing eyes turned upon us t..
    90769, ..e had a black beard, cut square at the end, and a pale face. I d..
    70631, .. no use for a beard save to conceal his features. Come in here, ..
   170308, ..e tall, black-bearded figure, his shoulders rounded, as he tipto..

>> pr.200 [249057]
   249057, ..eard!" <p> "A beard?" <p> "It is not the baronet -- it is -- why,
it is my neighbour, the convict!" <p> "With feverish haste we had turned the bo
dy over, and that dripping beard was pointing up to the..

>> stapleton near rock
  3: 2 matches

>> pr
   134931, ..d to see Miss Stapleton sitting upon a rock by the side of the t..
   246955, .. instant. And Stapleton, where is he? He shall answer for this d..

>> done
```

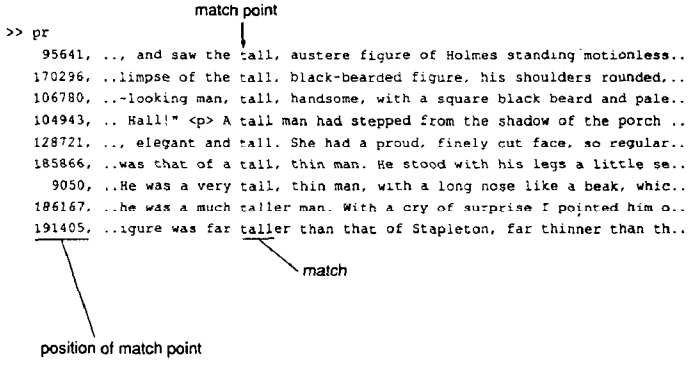
Figure 1. Example PAT session.

■ Your First Search

Searching



Printing



Your First Search

The first time you use PAT, the screen conventions will be unfamiliar. The facing page labels the important information on the screen.

prompt

The **>>** prompt means PAT is ready to accept a command. Typing a prefix, word, phrase, number or other text after the prompt and pressing the **Enter** or **Return** key starts the search, for example:

>> tall

search pattern

What you type is often referred to as a search pattern or *pattern* for short.

After you enter a pattern, PAT displays a line like the following:

1: 9 matches

set number

The number **1** is called the *set number*. It names a set of results with a number so you can use it in further searches. Following the set number is the result of the search. The number **9** is the number of times the pattern **tall** appears in the example text.

pr command

The **pr** command (short for print) shows one line of context around each occurrence of the search pattern, for example:

95641, .., and saw the tall, austere figure of Holmes standing motionless..

match point

The number in front stands for the position of the first character of the match (referred to as the *match point*). In the example, the letter **t** in **tall** is the 95,641st character in the text.

For each match, PAT prints two periods followed by 64 *characters* (14 to the left of the match point and 49 to the right) followed by two more periods. Note that spaces and punctuation as well as letters, numbers, and other symbols count as characters.

Figure 2. Layout of user manual.

searching session, subjects employed PAT to search a text and solve more difficult problems. The use of two sessions allowed us to evaluate the documentation separately from the software.

We selected Arthur Conan Doyle's *The Hound of the Baskervilles* as the text to be searched. This text is large enough to benefit from computer assistance for problem solving, but small enough to seem unimposing. Many people are acquainted with the main characters and intent of the document, but even those who might have read the story are unlikely to remember all its twists and turns. There is little overt structure other than chapters and paragraphs, and hence the document would not be cluttered with markup (as would have been the case with a dictionary or other reference text). Finally, we had ready availability to an online version of the text, and access to a local Doyle expert for advice. It was also appealing that our subjects would be, in effect, acting as detectives within a detective story.

The questions that comprised the searching task were chosen carefully with several criteria in mind. First, we wanted to engage the subjects' curiosity. Second, we wanted a range of difficulty, to ensure that the subjects could solve some of the queries, while being unlikely to solve all. Third, we wanted questions that would suggest the use of most of PAT's capabilities and test the limits of their understanding. Finally, we eschewed explicitly asking subjects to use a particular command to solve a given problem, partly because that seemed less realistic, and partly because their choice of technique would also be indicative of their knowledge of PAT.

2. METHOD.

Eighteen subjects participated in the experiment. Two were secretaries, five were library staff, and eleven were undergraduates at the University of Waterloo. We chose users who we thought would exhibit a wide range of experience in the use of searching systems and computers in general.

The experiment was conducted in two sessions. The first session was conducted with groups of between two and six subjects. Each subject was provided with a PAT manual, a ballpoint pen, a highlighter pen, blank paper, and a folder with the experimental material. The subjects first completed a simple questionnaire about their computer experience. Then the instructions for the remainder of the first session were read by the experimenter (the subjects also had typed copies of these instructions in their folders). The subjects were permitted to ask questions about the instructions or the experiment at any time.

In the main part of the first session, subjects familiarized themselves with PAT by reading the manual and attempting to answer ten PAT simulation problems. The problems required subjects to describe the input that would produce a given PAT output. Most of the problems could be answered by using the highlighter pen directly on the question page. The subjects were told that the problems would not be graded, and that they were intended solely to guide their reading to the sections of the manual that we thought would be the most useful. This part of the first session lasted for one hour.

In the final part of the first session, the experimenter discussed the problems. The correct answer for each problem was given, as well as an explanation of what the problem was intended to teach about PAT. Any questions raised by the subjects were answered fully by the experimenter, who tried to ensure that subjects had a complete and correct understanding of PAT's behaviour.

There was a one day gap between the first and second sessions.

The second session was conducted with pairs of subjects. The subjects were provided with all the material they had used in the first session. The subjects were introduced to one another if not already familiar, then the instructions for the second session were read by the experimenter. The subjects were provided with a Wyse 75 terminal, capable of displaying 24 lines by 80 characters. PAT was started before the subjects' arrival. After reading the instructions, the experimenter also told the subjects that he or she would be present during the session, behind a room divider, so that the subjects could be heard but not seen. The subjects were not told that their session was being recorded, or that the experimenter was observing their screen display on a slaved terminal.

In the main part of the second session, subjects used PAT to solve nine problems concerning the Arthur Conan Doyle story *The Hound of the Baskervilles*. The second session problems are shown in Figure 3. The experimenter did not interfere with the session unless the subjects inadvertently caused the terminal or PAT to suffer problems outside the bounds of the experiment. The subjects were encouraged to verbalize their problems and strategies as they searched.[5] The main part of the second session took one hour.

1. Find the number of times Holmes says *my dear Watson*.
2. Which characters have beards?
3. Which characters are referred to as handsome?
4. Does Miss Stapleton sit on a rock?
5. What brand of cigarette does Watson smoke?
6. See how much you can find out about Mr. Stapleton's physical features.
7. Which character is named most often in the book?
8. Which chapters include the phrase *I assure you*?
9. Who murdered whom?

Figure 3. Searching session problems.

In the final part of the second session, subjects were given the answers to the nine problems and were debriefed about the session. The subjects then completed a

questionnaire about the experiment, the documentation, and the software. Subjects were each paid twenty dollars for their participation.

3. RESULTS.

No data was collected on the subjects' answers to the problems in the first session, as we had informed the subjects that this session was not viewed as a test. Instead, we took note of the comments that subjects raised during the discussion period. Table 1 shows how many comments were made about different problems or uncertainties with PAT.

Comments	Commands				
	pattern	signif	docs	prox	total
Operation					
syntax	0	2	1	2	5
function	0	14	11	4	29
Match					
start	4	1	2	13	20
end	1	1	0	2	4

Table 1: Distribution of comments from Session 1.

The comments have been grouped by command, since subjects tended to identify problems according to commands rather than to specific questions. Examples of PAT's commands can be seen in Figure 4, which shows the quick reference card.† Of the nine simulation questions we provided, 2 questions dealt with searching for a pattern, 3 with `signif` (used for searching by frequency), 1 with `docs` (used for searching within restricted regions of the text), 2 with `shift` (used for manipulating the display), and 2 with proximity commands (`fby` and `near`). The comments for each command have been subdivided into two major categories: those that deal with the functionality of the command (its syntax or how it works), and those that deal with how the text was matched. In the latter case, we use the terms "start" and "end" to indicate when subjects had a comment about the starting position of a match (for example, would a search for "in" match the suffix of "within") and when they had a comment about the ending position of a match (for example, would a search for "in" match the word "inside").

Subjects had more difficulty with function than with syntax, and more difficulty with determining the start of a match than the end of a match. Subjects had many comments about the functionality of `signif` and `docs` but fewer problems with the text matched by `signif` and `docs`. Conversely, subjects had few problems with the functionality of the proximity commands but were confused about the positions of the matches. In particular, subjects thought it would be more natural to measure proximity as the shortest distance between any character of the two patterns, rather than measuring the distance between the starting point of both patterns.

The remainder of the tables present results collected from the searching session. Table 2 shows a summary of the pairs' usage of PAT commands. Since a single PAT command may consist of several parts, we counted each component separately and then summed them according to four categories: display of results (`pr, shift, sample`), proximity searching (`near, fby`), frequency searching (`signif`), and processing of restricted areas of the text (`docs, within, including`). The total number of command components is also given, not including errors or patterns.

Pair	Commands				
	display	prox	signif	docs	total
1	78%	16%	1%	4%	144
2	65	8	9	9	76
3	73	14	6	4	49
4	83	6	0	2	77
5	82	5	2	4	96
6	56	15	21	0	40
7	55	7	6	24	149
8	85	1	4	0	294
9	72	13	5	7	99

Table 2: Command Usage

It can be seen from Table 2 that the majority of subjects' effort was spent in displaying the search results, from a minimum of 55% of the command components to a maximum of 85%. The use of proximity commands was the next most frequent category, with the other categories amounting to less than 10% of the total except in two cases. The total number of command components ranged widely, from 40 to 294.

Table 3 shows the number of patterns used by the pairs during the whole session. "Patterns" are both explicit search strings (e.g., *whiskers*) and positional values entered by the subjects (in PAT one is allowed to specify a position in the text by stating its offset from the beginning of the text e.g., `[249057]`). Three values are given: the number of string patterns, the number of unique string patterns, and the number of positional patterns. The most surprising observation is that in 7 out of the 9 sessions 30% or more of the patterns are repetitions. PAT provides facilities for accessing previous results, so either subjects were not using these facilities, or found it easier just to re-enter searches, or were not confident of the answers that they had received and wanted to double-check.

Finally, Table 4 contains ratings of the subjects' effectiveness in solving the problems according to two methods. For method A, the pairs' performance in solving each of the questions was scored on a scale of 0 to 5, where 5 indicates complete solution of the problem and 0 indicates that the problem was not attempted. These values were summed and then expressed as a percentage of the

† Subjects did not have access to this card — its production was one of the results of the experiment.

What You Can Do		Examples
Access Pat	**Start and Stop Pat:**	
	Start Pat	`pat story`
	Leave Pat	`done` `quit` `stop`
Find Occurrences	**Find out how often something appears:**	
	A word	`"the "`
	Words that start as specified	`"the"`
	A phrase	`"to be or not to be "`
	A range of numbers or letters	`"10".."15"`
Print Context	**See some context around each match:**	
	One line of text	`pr`
	More characters to right	`pr.200`
	More characters on both sides	`pr.200 shift.-100`
	See some context around selected matches:	
	A specific match	`pr.500 [12345]`
	A specific set of matches	`pr 5`
	The previous set of matches	`pr %`
	A sample of 20 matches	`pr sample.20`
Search by Proximity	**Find text near to or far away from other text:**	
	A word near another (within 80 characters)	`war near peace`
	A word followed by another (within 100 characters)	`war fby.100 peace`
	A word not near another (not within 20 characters)	`war not near.20 peace`
	A word not followed by another (not within 80 characters)	`war not fby peace`
Search by Frequency	**Find text that appears often:**	
	The most frequent word or phrase	`signif ""`
	...that starts with `green`	`signif "green "`
	The 10 most frequent words or phrases	`signif.-10 ""`
	...that start with `upon`	`signif.-10 "upon "`
	The most frequent three-word phrase	`signif.3 ""`
	... that starts with `the`	`signif.3 "the "`
	The longest repeated phrase(s)	`lrep ""`
	...that starts with `one`	`lrep "one "`
	...that are longer than 20 characters	`lrep.20 "one "`
Restrict Searching Area	**Find text within a pre-defined area:**	
	Find `moor` within chapters	`moor within docs chap`
	Find start of chapter(s) containing `moor`	`docs chap including moor`
	...that contain 5 or more references	`docs chap including.5 moor`
	Print to end of chapter	`pr.docs.chap`
	Create your own area to search	
	Define paragraph components	`para = docs "<p>".."</p>"`
	Find `hound` within paragraphs	`"hound " within *para`
	Find start of paragraphs containing `hound`	`*para including "hound "`
	Print to end of paragraph	`pr.docs.*para`

Figure 4: Quick Reference Guide to PAT.

Pair	Patterns		
	alphabetic		positional
	total	unique	
1	123	53	5
2	91	63	17
3	39	31	2
4	59	53	31
5	77	42	18
6	74	41	0
7	148	63	2
8	95	44	2
9	79	40	3

Table 3: Pattern Usage

maximum possible score. Method A treats all questions as having equivalent value, and involves some subjectivity about how much of the question was completed. For method B, the questions were weighted according to the number of distinct facts that were considered necessary to solve the problem, and then expressed as a percentage of the maximum possible score. In method B, determining the brand of cigarette that Watson smokes counts for only 5 percent of the total score. while determining Mr. Stapleton's physical features (light hair, grey eyes, prim-faced, lean-jawed, between thirty and forty years old) counts for 25 percent of the total score. Six of the nine pairs performed well according to method A (i.e., achieved part of the answer to the majority of the questions), but only two pairs maintained a high rating under method B (i.e., completed a majority of the work).

Pair	Solutions	
	A	B
1	60%	25%
2	71	40
3	27	15
4	89	80
5	64	35
6	11	5
7	36	20
8	82	70
9	56	25

Table 4: Effectiveness

Regression analysis was performed using Perlman's ISTAT package.[6] The number of display commands used was correlated to method B solution values ($F(1,7)=6.54$, $p=0.037$) and the number of proximity commands used was inversely correlated to the method B solution values ($F(1,7)=8.82$, $p=0.021$). Hence better performance was correlated with greater use of display functions, but somewhat surprisingly, was correlated with lesser use of proximity functions. We noticed that group 6 had a significant influence on this latter result, since their performance was the lowest and their use of proximity functions was the highest. Elimination of this group from the analysis still results in a marginally significant finding for correlation of proximity and method B solution value ($F(1,6)=5.53$, $p=0.057$). No other correlations were detected.

At the end of session 2, subjects answered a questionnaire about the tasks they performed, PAT, and the documentation. Not all subjects answered all questions, partly because some pairs did not attempt all the commands. 14 of 18 subjects rated the tasks as above average in difficulty. 7 subjects said they were most successful at question 1 ("my dear watson"). 4 subjects did not indicate a specific task, but did indicate that they felt most successful with simple searches. When asked which tasks they felt least successful in solving, 9 subjects chose the "beard", "handsome" and "murder" problems. 4 other subjects indicated problems of this type by giving answers such as "those involving context searching."

The second part of the questionnaire asked subjects to evaluate PAT. Not surprisingly, the `signif` and `docs` commands were considered the hardest commands. `signif` was rated as above average in difficulty by 7 out of 16 subjects; `docs` was rated as above average in difficulty by 8 out of 15 subjects.

The rating of the documentation followed the same trend, with 7 out of 17 subjects rating the explanation of `signif` as above average in difficulty and 3 out of 16 rating `docs` the same. The reason the documentation of `docs` fared considerably better than the rating for the command itself may be that the experimental problem was quite similar to an example in the documentation.

4. DISCUSSION.

The results of command usage show clearly that seeing an inch of document (or in the case of PAT, 65 characters) of context around a match is not sufficient except in the simplest of cases. The majority of subjects' commands to PAT were to display text. Furthermore, the subjects who expended more effort on display generally did better in finding results — whereas, by contrast, we did not observe that those subjects who used more search patterns or who used more of PAT's features exhibited better performance.

These results suggest that improving display capabilities will reduce effort while keeping performance high. Consider, for example, that each search result in PAT requires an explicit display command, and most search commands are followed by a display request. A large fraction of this effort could be avoided if PAT's default were to print a sample of the results. Another problem is the low-level nature of PAT's display operations, requiring that the user specify an absolute position and a range of characters to be displayed, as is shown in Figure 1. Apart from being tedious to use, number-based specifications were confused by our subjects with the numbers that occur in the text, the numbers used as parameters to PAT commands, and the numbers that are assigned to the results of previous queries. Avoiding this type of conflict is a prime requirement of an

improved display system.

A more subtle indication of the need for improved display arises from the problems *Who murdered whom?* and *Find Stapleton's physical features.* These were the most difficult problems the subjects had to solve, partly because the answers could not be found in one section of the text†. Even where the answers are given, long stretches of text separate the description of the event or person and the mention of a name. Furthermore, common structural cues such as sentences and paragraphs were not directly available for search or display. These problems meant that it was more difficult for subjects to acquire reasonable evidence quickly, and so they tended to give up and move on to some other clue.

The second major problem we noticed was that subjects were not clear about the distinction between lexical and semantic searching, nor were they aware of the separate roles of the document and the index in determining what could be found. In solving the query *Which characters have beards?*, for example, some pairs would enter *beards*; since the plural of "beard" does not appear in the story, they decided that none of the characters had beards. The following set of queries also provides interesting evidence of the mistaken notion that PAT searches semantically:

```
>> docs chap including characters with beards
>>  ("beards" on characters) within docs chap
>> ("beard" of characters) within docs chap
>> " bearded characters " within docs chap
>> *chap including ("beards" on characters)
```

Each of these queries is syntactically faulty; however, the important observation is that the subjects are showing their confusion about the distinction between lexical and semantic searching. The suggestive connotations of the command variables `including` and `within` has led subjects to suppose that PAT understands semantic relationships, such as the relationship between people and beards. It has also suggested the use of other prepositions like "on", "of", and "with", which seem more reasonable descriptions of the relationship between people and beards than "within" or "including."

Further evidence of the confusion between semantic and lexical searching is provided by the varieties of patterns submitted by the subjects. For example, to solve the beard problem, subjects tried *beard* (18 occurrences), *beards* (0 occurrences), *bearded* (3 occurrences), *facial hair* (0 occurrences), and *hairy* (1 occurrence). *bearded* found no new evidence because it is prefixed by *beard*, and *hairy* does not refer to a character in the story. What is interesting about these words is that they seem to be unlikely lexical variants. Subjects appear to be treating PAT as if it were a system for looking up keywords; that is, they chose words that were synonymous without considering whether they were likely to appear in the text.

A last important observation involves the comparison of subjects' problems in the two sessions. In the first session subjects had problems with understanding the concept of `signif`. Considering its multiple forms, non-intuitive syntax, and rather foreign functionality of `signif`, confusion is not surprising. We counted at least eight different misinterpretations of `signif`. Perhaps its difficulty caused subjects to focus on `signif`, as 4 of the 10 pairs considered its use in the second session to find the number of occurrences of *my dear watson.* 3 of the pairs actually entered the query `signif my dear watson`. That subjects should attempt to solve the simplest problem with the most complicated of PAT commands is less a difficulty with `signif` than an indication of the subjects' misunderstanding of the basic functionality of PAT.

Similarly, the first session suggested that subjects had considerable difficulty with understanding the limits of matching for the proximity functions. These difficulties did not surface during the searching session, possibly because precision in proximity was not required. Subjects appeared to be comfortable with the notion of proximity in the training session. Their use of it in the second session, however, was correlated with poorer performance. A possible explanation is that proximity-based functions were diverting them from more productive activity. The results related to `signif` and the proximity functions show that the training session was exposing problems other than those that showed up in the searching session. Hence the experimental methodology provided feedback that would not have been obtained if we had combined the two sessions.

Both sessions exposed a large number of problems with the specifics of both PAT and the documentation. For example, the command `pr.100` prints 100 characters of text to the right of the match; subjects issued the command `pr.-100`, hoping for text to be displayed to the left of the match. This extrapolation, although syntactically invalid, was perfectly reasonable since other commands in PAT have signed parameters. The design of the system should accommodate such extrapolations.[7] Similarly, the experiment provided feedback on the flaws and inadequacies in the user manual. Perhaps the most obvious of these was the confusing phrase "character sequence" which was employed to describe text being matched or used as a search pattern. This terminology contributed to the confusion about whether PAT matches the start of words and also the middle of words; subjects thought that the phrase "character sequence" meant the latter. Another problematic term was "docs", a word used both as a short form

† Stapleton murders Baskerville and Selden, and attempts another murder. However, several facets of the story lead to confusion: the hound does the actual killing; Stapleton is also a Baskerville, unbeknownst to the other characters; Stapleton himself thinks that he has killed Sir Charles, when really he has killed Selden in Sir Charles's clothing; the main murder takes place chronologically before the events that make up the text of the story.

Stapleton's physical features are described in several places, including when he is in disguise as a spy in a cab (the cabman thinks he is Sherlock Holmes) and when he appears in a painting on a wall of Baskerville Hall.

for "documents", (subcomponents of the text) and as part of two PAT commands that employ subcomponents. Some subjects thought "docs" meant a text file, as opposed to subcomponents of the text. Although the PAT syntax was not altered, the manual was revised to use the term "text component."

5. IMPLICATIONS.

Online documentation can be searched with full text tools in much the same way as *The Hound of the Baskervilles*. In both situations, users are looking for just enough information to answer a question or confirm what they already suspect. This type of searching is quite different from traditional library searching, where the goal is retrieval of all information relevant to a query. Therefore some of the problems we have described and results we have obtained will be more useful in addressing full text systems for online documentation than will previous research in library searching.

We found that users have some difficulty with both the concepts and the syntax of PAT. Documentors must ensure that users understand the difference between searching for lexical strings and searching for semantic categories, especially since users are more likely to be familiar with the latter. While it is simple to introduce users to full text search by means of examples, it will ultimately be necessary to explain why and how full text searching works, and why it can fail to provide answers. Every document will differ on many aspects that affect even the simplest search; for example, which points of the text are indexed, which words or characters are ignored, the case or punctuation-sensitivity, and which subcomponents are defined. Similarly, the particular searching software has its own characteristics; for example, morphological support, the interaction of a query with the current session, and the treatment of queries as prefixes, suffixes, whole words, or phrases. All these issues interact in a complex fashion that results in an environment seen by the user as "the system." It is interesting to note that differentiating between these issues is seen by the novice as unnecessary complexity, though the serious user must regard them as essential for effective use of the software.

It is also important that users have access to good context display tools so they can navigate around their matches. At the Centre for the New Oxford English Dictionary, we have built a context display tool to address this problem. Users now take advantage of the powerful searching capabilities of PAT, but leave the context display to LECTOR, a tool for flexible display of tagged text. [8] Multiple invocations of LECTOR can be used to provide several simultaneous views of a text. Figure 5 shows PAT and LECTOR being used to search the online version of the user manual. Each LECTOR window provides a different context, suppressing various parts of the manual and varying the formatting. Thus in addition to displaying the user guide in its entirety, sections of the Guide have been exposed (based on underlying tags) to create other views of the text. Figure 5 shows a display of the headings, a display of the example commands, and a display of the glossary terms. Where the match is visible, it is highlighted.

We found our experimental method of testing the documentation in isolation provided us with several benefits. First, we could trace documentation problems directly to the documentation. For example, the problem subjects had with determining whether PAT was searching for words or characters was largely the result of inappropriate terminology in the manual. Second, we could direct the user to the parts of the manual that we thought needed the most attention. By forcing users to rely exclusively on the documentation without the benefit of trial and error use of the system, we identified places where the documentation was incomplete or inadequate. When the information was incomplete or not comprehended, subjects relied on their intuition. Their comments provided us with input on how they expected the system to work. This method may be advantageous in the early stages of software and documentation development, when a paper prototype could be used to obtain feedback for the design of the user interface and functionality of the system.[9]

The experimental method also had certain costs. First, it required subjects to attend two sessions. Second, as a training method it proved inadequate and perhaps more confusing than permitting subjects to use the software immediately. Despite the hour-long session with the documentation and the followup discussion, many subjects still had problems with the basic concepts and functions of PAT. One pair of subjects still had not grasped the notion of searching for lexical strings in the text. As a result, we cannot recommend use of this strategy for training.

6. CONCLUSIONS.

Full text systems can be extremely useful for searching online text, particularly when the searching problem is a fact-finding one rather than one of retrieving all relevant documents. Empirical evidence suggests that users are more effective when they can see more of the text, so it is important to provide good display tools. We did not find that creativity or the use of more esoteric searching features provided better results. The documentation of full text systems is complicated by the strong interaction between document, index, and software.

7. ACKNOWLEDGEMENTS.

Our thanks to Frank R. Safayeni and his students for help in designing the experiment and the questionnaires; to Paul Beam for providing experimental subjects; to Chris Redmond for sharing his knowledge and enthusiasm for Holmes; and to Edmund Weiner for his suggestions during the writing of this report. We are also grateful for the financial support of the Natural Science and Engineering Research Council of Canada under University-Industry grant 0039063.

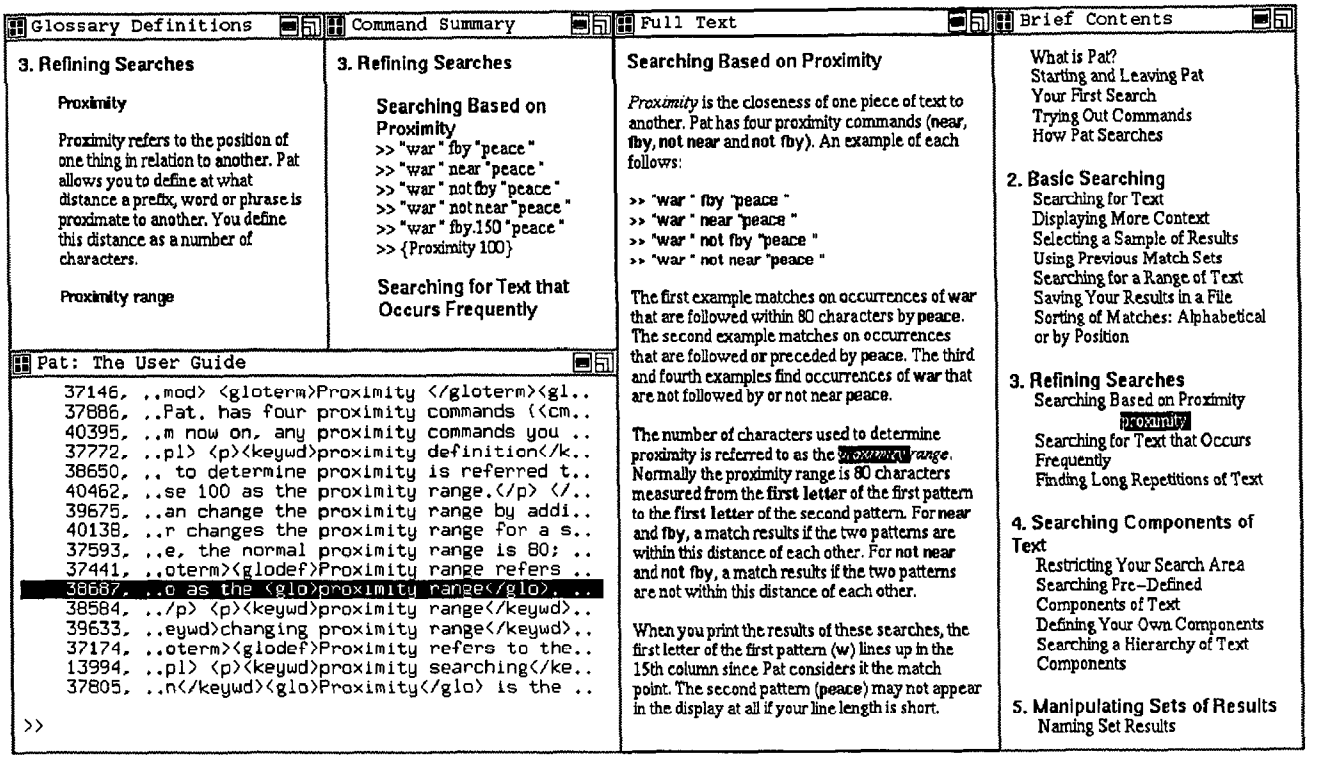


Figure 5: Multi-LECTOR view of online PAT manual.