2000

On Pattern-Directed Search of Archives and Collections

Garett Dworman
University of Pennsylvania

Steven. O. Kimbrough
University of Pennsylvania

Chuck Patch

Recommended Citation

Dworman, G., Kimbrough, S. O., & Patch, C. (2000). On Pattern-Directed Search of Archives and Collections. *Journal of the Association for Information Science and Technology, 51* (1), 14-23. http://dx.doi.org/10.1002/(SICI)1097-4571(2000)51:1<14::AID-ASI4>3.0.CO;2-N

On Pattern-Directed Search of Archives and Collections

Abstract

This article begins by presenting and discussing the distinction between record-oriented and pattern-oriented search. Examples of record-oriented (or item-oriented) questions include: “What (or how many, etc.) glass items made prior to 100 A.D. do we have in our collection?” and “How many paintings featuring dogs do we have that were painted during the 19th century, and who painted them?” Standard database systems are well suited to answering such questions, based on the data in, for example, a collections management system. Examples of pattern-oriented questions include: “How does the (apparent) production of glass objects vary over time between 400 B.C. and 100 A.D.?” and “What other animals are present in paintings with dogs (painted during the 19th century and in our collection)?” Standard database systems are not well suited to answering these sorts of questions (and pattern-oriented questions in general), even though the basic data is properly stored in them. To answer pattern-oriented questions it is the accepted solution to transform the underlying (relational) data to what is called the data cube or cross tabulation form (there are other forms as well). We discuss how this can be done for non-numeric data, such as are found widely in museum collections and archives. Further we discuss and demonstrate two distinct, but related, approaches to exploring for patterns in such cross tabulated museum data. The two approaches have been implemented as the prototype systems Homer and MOTC. We conclude by discussing initial experimental evidence indicating that these approaches are indeed effective in helping people find answers to their pattern-oriented questions of museum and archive collections.

Disciplines

Other Arts and Humanities | Other Education | Painting

Pattern-Directed Search of Archives and Collections[†]

Garett O. Dworman, Steven O. Kimbrough, Chuck Patch

Abstract:

This paper begins by presenting and discussing the distinction between record-oriented and pattern-oriented search. Examples of record-oriented (or item-oriented) questions include: "What (or how many, etc.) glass items made prior to 100 AD do we have in our collection?" and "How many paintings featuring dogs do we have that were painted during the 19th century, and who painted them?". Standard database systems are well suited to answering such questions, based on the data in, e.g., a collections management system. Examples of pattern-oriented questions include: "How does the (apparent) production of glass objects vary over time between 400 BC and 100 AD?" and "What other animals are present in paintings with dogs (painted during the 19th century and in our collection)?". Standard database systems are not well suited to answering these sorts of questions (and pattern-oriented questions in general), even though the basic data is properly stored in them. To answer pattern-oriented questions it is the accepted solution to transform the underlying (relational) data to what is called the data cube or cross tabulation form (there are other forms as well). We discuss how this can be done for non-numeric data, such as are found widely in museum collections and archives. Further we discuss and demonstrate two distinct, but related, approaches to exploring for patterns in such cross tabulated museum data. The two approaches have been implemented as the prototype systems Homer and MOTC. We conclude by discussing initial experimental evidence indicating that these approaches are indeed effective in helping people find answers to their pattern-oriented questions of museum and archive collections.

Authors:

Garett Dworman is a Ph.D. candidate in the Department of Operations and Information Management at The Wharton School of the University of Pennsylvania. The main theme of his research is the design of cognitively motivated information access systems. For his dissertation he is developing pattern-oriented systems for accessing document collections. This technology is currently being applied to collections in museums and the health-care industry. Address: University of Pennsylvania, 3620 Locust Walk, Suite 1300, Philadelphia, PA 19104-6366. Tel: (215) 898-5133. Fax: (215) 898-3664. Email: dworman@opim.wharton.upenn.edu. URL: http://opim.wharton.upenn.edu/~dworman/ .

Steven O. Kimbrough is a Professor at The Wharton School, University of Pennsylvania. He received his Ph.D. in philosophy from the University of Wisconsin. His active research areas are: formal languages for business communication, evolutionary computation (including genetic algorithms and genetic programming), decision support systems, and information mining and retrieval. He is currently co-Principal Investigator of the Logistics DSS project, which is part of DARPA's Advanced Logistics Program. Address: University of Pennsylvania, 3620 Locust Walk, Suite 1300, Philadelphia, PA 19104-6366. Tel: (215) 898-5133. Fax: (215) 898-3664. Email: kimbrough@wharton.upenn.edu. URL: http://grace.wharton.upenn.edu/~sok/ .

Chuck Patch is the Director of Systems at the Historic New Orleans Collection. He is responsible for all automation projects at his institution. He can be reached at The Historic New Orleans Collection, 533 Royal Street, New Orleans, LA 70130. (504) 523-4662. Fax: (504) 598-7108. Email chuckp@hnoc.org. URL: http://www.hnoc.org/ .

[†] File: mw99-19990128.doc.

Two Kinds of Questions

One’s purpose, when approaching an archive or museum collection for information, might be characterized as seeking an answer to one or more questions. Thus, if an information system is to be helpful in answering one’s questions of archives and collections, it would seen that categorizing the questions to be asked can only be helpful in designing an information system to assist in answering them. What kinds of questions are there that are pertinent to archives and museum collections? This is a large and difficult issue, and we do not essay to resolve it here. Our aim in this paper is more modest: we wish to distinguish two kinds of questions and to explore their relevance to museum and archive informatics. We devote the remainder of the present section to making and exploring our basic distinction. The sections that follow explore the distinction in the context of a particular information system, the Core of Discovery system, installed at The Historic New Orleans Collection.

The distinction we wish to make here, and to exploit in designing museum and archive information systems, is deeply embedded in folklore and ordinary language. "You cannot see the wood for the trees" is perhaps the earliest recorded embodiment of the distinction in English. (The quotation is from John Heywood’s *Proverbs*, itself the earliest published (1546) collection of English folk sayings.) Proverbially, there is a distinction to be made between seeing (or asking about) the trees and seeing (or asking about) the forest. But how can we characterize the distinction and what can we do to provide computerized support for these two kinds of questions? One question at a time. First, a characterization of the distinction.

The distinction is best seen through a series of examples. Let us compare some tree questions with some forest questions. Here are some questions about trees in a forest.

1) What kind of tree is this?
2) Which are the birch trees?
3) Which conifers are less than five years old?
4) How many oak trees are there?

The reader can no doubt think of many other examples. Simply imagine that we have a catalog of a forest with a record for each (individual) tree. These individual tree records record what we know about each of the trees. These records contain the answers to a great many questions we might want to ask. Such questions are typically about the attributes of a given tree, or type of tree. They typically request either a display of records (individual tree records) satisfying a certain condition (e.g., questions 1, 2 and 3, above), or a numerical summary of records satisfying a certain condition (e.g., question 4, above). Such questions might be called *trees questions*; we prefer the more directly suggestive *record-oriented questions*.

The records that record-oriented questions address and seek information about are, when computerized, usually either database records or individual text records. (Of course, records may be other things as well. They may be paper note cards in a file. They may be digitized images, movies or sound recordings stored on disk. Computerized access methods are, however, most developed for database records and texts, so we focus the discussion on these.) Operationally, there is a quite precise way of characterizing record-oriented questions for database records: these are the questions that may be asked of a database using the SQL SELECT statement. Question 3, above, might be symbolized into SQL as

```
SELECT  *
FROM Trees
WHERE (Trees.Type='conifer' AND Trees.Age<=5);
```

Question 4 might be rendered into SQL as

```
SELECT COUNT(*)
FROM Trees
WHERE (Trees.Type='oak');
```

If, as is often the case, the available records are not in database format, but are texts, the problem of answering record-oriented questions is much more challenging. Database systems and SQL are not the primary tools; information retrieval systems are (e.g., Blair, 1990; Korfhage, 1997; Salton & McGill, 1983; van Rijsbergen, 1979). Now we are generally in the situation of trying to find particular documents, or texts, containing the information that answers our question. To do this, we guess at search terms or combinations of search terms and ask our information retrieval systems to present us a list of documents (records in our broader sense) matching the query terms. We can then peruse the returned records and hope either to find the answer to our question or to obtain information useful for refining our query.

All this is well and good, but what about the forest? Here are some questions about a forest.

5) How does the mixture of tree types vary by distance from a stream or other form of surface water?

6) Do the different varieties of conifer prosper differentially by soil acidity?

7) Are there more deciduous trees at greater heights?

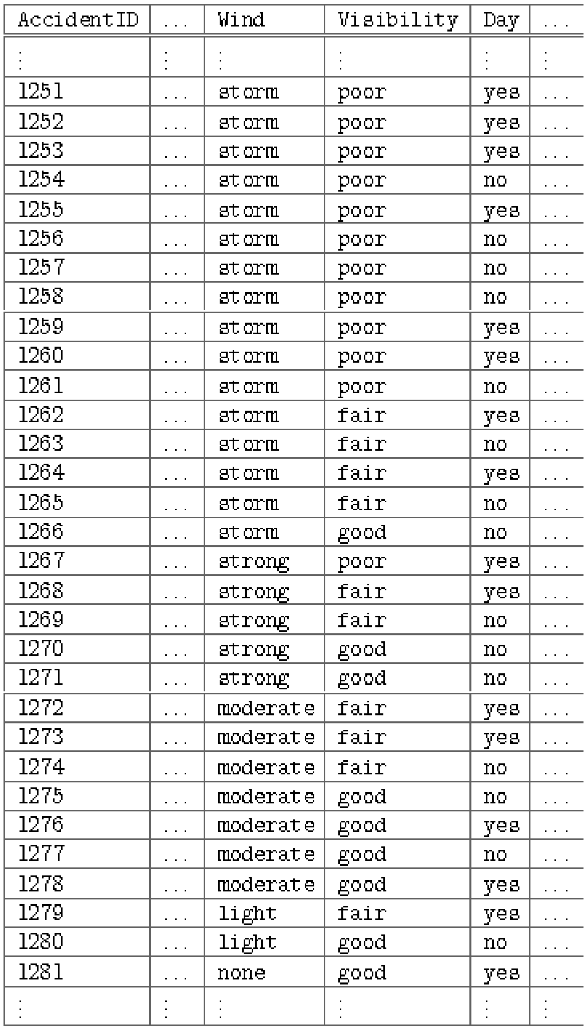
8) Do the older trees that are on hillsides tend to have a fire-resistant type of bark?

We trust the reader will recognize these as entirely valid, and often-asked, types of questions. How do they differ from questions 1-4, above, the record-oriented questions? The fundamental difference that we see is this. Record-oriented questions ask about *one* type of thing: birch trees, conifers less than five years old, oak trees, and so on. Forest, or as we call them *pattern-oriented*, questions ask about *two or more* kinds of thing. They ask about *relationships* between and among things: (5) What is the relationship between *tree type* and *distance from water*? (6) What is the relationship *between frequency of conifer types* and *degrees of soil acidity*? (8) What is the relationship between *tree age*, *terrain location of trees*, and *type of bark*? And so on. (There are questions of a pattern-oriented nature for which SQL SELECT is adequate: What kinds of trees are there in the forest and how frequently does each kind appear? or Which kind of tree occurs most frequently? Notice, however, that these are limiting cases, in which really only one variable (with different values) is under consideration.)

Typically for pattern-oriented questions, we have a series of variables (X, Y, Z, …) for types of things (conifer types trees, trees located in various types of terrain, etc.) and we are asking for associations among them. If X is high (conifer type 4 or 5) and Z is middling (2, 3 or 4), does Y tend to be low (terrain type 1 or 2)? (Here the numerical coding is only for convenience. Relationships may usefully be studied among ordinal or even nominal variables.) Because pattern-oriented questions are not about a single type of thing, there is not a single type of record that can answer to them. No single record-oriented query can answer a pattern-oriented question; the answer to a pattern-oriented query resides in the patterns among the records, not in any individual record itself. (Of course, in the information retrieval context it is conceivable that we might get lucky and retrieve a document that happened to answer our pattern-oriented question, but this eventuality can be neglected.)

What to do? How, if the SQL SELECT statement won't work, can we possibly support record-oriented questions with an information system? One thing we can do is to transform or re-represent the records we do have in order to facilitate pattern-oriented queries. This is exactly what is done with relational database records for purposes of database mining. Properly normalized relational databases are denormalized in special ways in order to make it easier to get answers to pattern-oriented ("slicing and dicing") questions. For this purpose, the database mining world recognizes the "data cube" or "multidimensional" form, which is really a simple kind of cross tabulation of underlying records. A simple example should help make the concepts clearer. See Figure 1, which shows in schematic form a series of database records concerning boating accidents (the data are hypothetical but realistic).

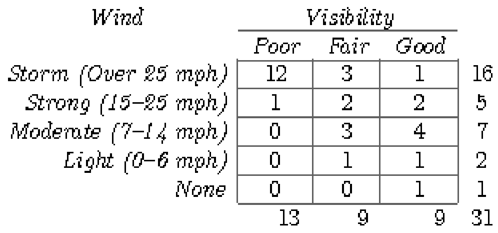
Figure 1: (Notional) Records Pertaining to Boating Accidents



AccidentID	...	Wind	Visibility	Day	...
⋮	⋮	⋮	⋮	⋮	⋮
1251	...	storm	poor	yes	...
1252	...	storm	poor	yes	...
1253	...	storm	poor	yes	...
1254	...	storm	poor	no	...
1255	...	storm	poor	yes	...
1256	...	storm	poor	no	...
1257	...	storm	poor	no	...
1258	...	storm	poor	no	...
1259	...	storm	poor	yes	...
1260	...	storm	poor	yes	...
1261	...	storm	poor	no	...
1262	...	storm	fair	yes	...
1263	...	storm	fair	no	...
1264	...	storm	fair	yes	...
1265	...	storm	fair	no	...
1266	...	storm	good	no	...
1267	...	strong	poor	yes	...
1268	...	strong	fair	yes	...
1269	...	strong	fair	no	...
1270	...	strong	good	no	...
1271	...	strong	good	no	...
1272	...	moderate	fair	yes	...
1273	...	moderate	fair	yes	...
1274	...	moderate	fair	no	...
1275	...	moderate	good	no	...
1276	...	moderate	good	yes	...
1277	...	moderate	good	no	...
1278	...	moderate	good	yes	...
1279	...	light	fair	yes	...
1280	...	light	good	no	...
1281	...	none	good	yes	...
⋮	⋮	⋮	⋮	⋮	⋮

Suppose now we are interested in understanding how visibility and wind conditions interact in association with boating accidents (causation is another matter). All the information we have is in the records recorded in Figure 1, but it is difficult to see or to extract automatically the patterns of association among these (or any other) variables. Figure 2, however, shows the cross tabulation of wind and visibility, and the nature of the association is now rather plain.

Figure 2: Cross Tabulation of Wind and Visibility Information in Accident Data Records



Wind	Visibility			
	Poor	Fair	Good	
Storm (Over 25 mph)	12	3	1	16
Strong (15–25 mph)	1	2	2	5
Moderate (7–14 mph)	0	3	4	7
Light (0–6 mph)	0	1	1	2
None	0	0	1	1
	13	9	9	31

Of the 31 accident records, there are 12 cases in which visibility was poor and storm conditions present, 3 cases in which visibility was fair and storm conditions present, and so on. Because of data aggregation, Figure 2 actually contains less information than Figure 1, but for purposes of pattern-oriented questioning, it is much more immediately useful. Experience has shown this and related forms to be amenable to recognizing patterns both visually and by programs. Moreover, the strategy of taking a cross tabulation generalizes to many dimensions, although using more than 5 or 6 at once is rare. (Space limitations prevent us from providing a more complete account, but the idea is a standard one and there is much available written on it. See, e.g., for additional details Balachandran et al., 1999; Codd et al., 1993; Dhar & Stein, 1997; and Hildebrand et al. Also, in Microsoft's Excel spreadsheet product, there is useful online help available. Search on "pivot table.")

There is now a substantial literature, and even an industry, devoted to transforming relational database data into crosstab forms so that pattern-oriented queries can be processed with acceptable efficiency. What about pattern-oriented questions directed at collections of textual records? Perhaps surprisingly, there is very little literature and there is certainly no industry. (The standard sources on information retrieval, such as those cited above, say very little or nothing about the problem of pattern-oriented retrieval of information in textual documents.)

The literature that does exist on pattern-oriented querying of collections of text is intriguing, but very thin. Don Swanson has made the most notable contributions. He has discovered a number of plausible hypotheses in the medical literature, using word count data and other standard information retrieval techniques, along with considerable ingenuity and diligence. To cite one of several examples, Swanson (1988) hypothesized a relationship between magnesium levels and migraine headaches based upon his studies of the literature in MEDLINE. Specifically, he hypothesized that magnesium deficiencies may cause migraine attacks. He based the hypothesis on his discovery of pairs of articles with related titles such as the following:

- "The relation of migraine and epilepsy" and "The magnesium deficient rat as a model of epilepsy"
- "Role of calcium entry blockers in the prophylaxis of migraine" and "Magnesium: nature's physiologic calcium blocker"

Swanson's 1988 study cites 128 articles containing 11 different intermediate topics, such as *epilepsy*, linking migraines and magnesium. Yet there was no mention in any MEDLINE document of any relationship between the two.

> Remarkably, none of the sixty-five articles on migraine mentions or cites any articles on magnesium and none of the sixty-three articles on magnesium mentions or cites any articles on migraine. Moreover, among 4,600 migraine records and 38,000 magnesium records, there were only six that contained both "migraine" and "magnesium." The six corresponding articles, published over a twenty year time span, were principally on magnesium. They offered little or no substantive discussion of the migraine literature and none had been cited by any migraine researcher, as judged by searching the Science Citation Index. In short, neither online searching nor printed indexes nor reading the text and following citation trails in medical articles turned up evidence that there was, at the time, any substantial scientific interest in the possibility of a physiological relationship between magnesium and migraine. (Swanson, 1993)

The hypothesis has since been confirmed. This example, and a few others (mainly from Swanson), demonstrates the potential value of searching for patterns in collections of text. Is there anything that can be done, analogous to what is done in database mining, to support pattern-oriented queries with information system? There is, and that story begins in the next section.

The Core of Discovery System

The Core of Discovery is a prototype system for exploring collections of textual data. It is currently installed and in use at The Historic New Orleans Collection, operating on the archives of the photographer Clarence John Laughlin. Laughlin took more than 15,000 photographs between 1930 and 1975. Remarkably, he wrote short comments on most of his photographs.

> Laughlin was fond of saying that he was a writer first, a book collector second and a photographer third. While he was undoubtedly being intentionally provocative, he also sincerely believed his photography to be merely an outgrowth, or another expression of his innate interests in poetry, philosophy, architecture, and the symbolic uses of objects. He took an almost synesthetic stance toward his work---referring to many of his photographs as visual poems. He was adamant throughout his career about including his long and elaborate captions on the walls of the exhibits and on the pages of his books---insisting that they were equal in importance to the images. (Patch, 1994)

The Core of Discovery system indexes Laughlin's comments on his photos and integrates the resulting indices with data about the photographs stored in The Historic New Orleans Collection's collections management system. With this extended indexing available, the Core of Discovery system offers three distinct retrieval services.

a) **Keyword retrieval**. The Keyword retrieval service is a simple term matching mechanism with no relevance ranking. The purpose of this module is to allow a user to locate photographs by specifying terms in the titles or captions of the desired photographs. This (record-oriented) retrieval service, while necessary, is entirely standard and present in nearly all systems. We shall have nothing further to say about it here.

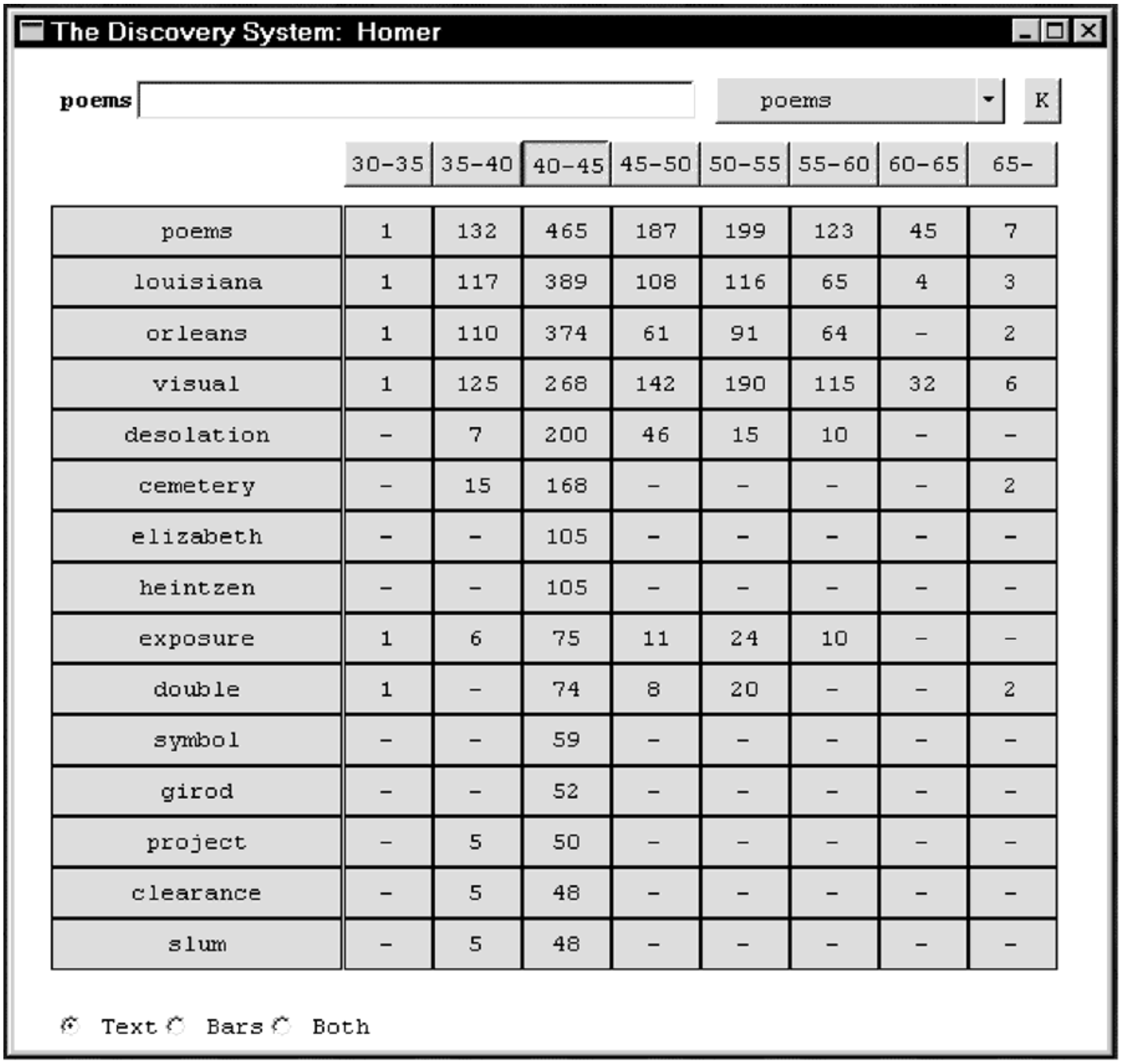
b) **Concept retrieval**. The concept retrieval service uses a ranking algorithm called DCB to rank photographs by relevance to a specified topic. (Laughlin's text about the photographs is used by the DCB algorithm to create the rankings.) The purpose of this service is to allow a user to find photographs that appear to be about a given topic, whether or not the keyword identified by the user appears in Laughlin's description of the photograph. Details regarding the DCB ranking algorithm, including experimental evidence that indicates very effective performance, may be found in (Dworman et al., 1997).

c) **Pattern-oriented retrieval**. The pattern-oriented retrieval service called Homer displays global information about the Laughlin collection so that users may find trends and associations among the collection topics. It is unique or nearly so (as far as we know) in providing fully automated and interactive support for pattern-oriented queries directed at collections of texts.

In what follows, we focus on Homer, the pattern-oriented retrieval service in the Core of Discovery.

Homer (Dworman, 1996; Dworman, 1998) is a generic system for finding and viewing patterns in collections of text. In the Core of Discovery system presently installed at The Historic New Orleans Collection, Homer is configured in a specialized fashion. Although we will discuss it in that context, the reader should understand that we do this for the sake of concreteness. Homer is quite general-purpose and has been applied successfully to many different data sets. In order to see what Homer does consider Figure 3, which presents Homer's main (and for our purposes, only) screen.

Figure 3: Homer Display



	30-35	35-40	40-45	45-50	50-55	55-60	60-65	65-
poems	1	132	465	187	199	123	45	7
louisiana	1	117	389	108	116	65	4	3
orleans	1	110	374	61	91	64	–	2
visual	1	125	268	142	190	115	32	6
desolation	–	7	200	46	15	10	–	–
cemetery	–	15	168	–	–	–	–	2
elizabeth	–	–	105	–	–	–	–	–
heintzen	–	–	105	–	–	–	–	–
exposure	1	6	75	11	24	10	–	–
double	1	–	74	8	20	–	–	2
symbol	–	–	59	–	–	–	–	–
girod	–	–	52	–	–	–	–	–
project	–	5	50	–	–	–	–	–
clearance	–	5	48	–	–	–	–	–
slum	–	5	48	–	–	–	–	–

● Text ○ Bars ○ Both

Here is how this display is to be interpreted. We are looking at information derived from the records of the Laughlin archives, including especially the texts he created to describe his various photographs. On the left-hand side of this display we see a column of words that appear in Laughlin's descriptions of his photographs: "poems", "louisiana", "orleans" and so on. Across the top of the display the columns are labeled with time intervals: 30-35, for example, indicates the years 1930 through the end of 1934, 35-40, the years 1935 through the end of 1939, and so on. Above the column headings we see the word poems displayed, indicating that the user has told Homer to look for patterns associated with the word poems.

Homer has then produced the display Figure 3. Further, although it is a bit difficult to see in this rendering of the interface (and easy to see with the real system), the user has selected the column headed 40-45 (1940-1944). Homer has sorted the terms (for that column) in descending order by frequency of occurrence. Thus, of all the Laughlin photographs dated in the 1940-1944 interval and containing the word poems, the word poems occurs more often in the Laughlin records (text he wrote about the photographs) than any other word (excluding various "stop" words, such as "and", "or", and "the"). In fact, it occurs 465 times. (It is not in the least surprising that the word poems occurs most often in documents containing the word poems. Absent ties, this has to be true. Also, we note that a later version of Homer displays the total number of photographs in the time interval, but this version is not currently installed in New Orleans.) Next after the word poems, the word louisiana occurs the most often, 389 times. And so on down the column. By clicking on a different column heading (corresponding to a different five-year period), the user can direct Homer to sort the column in question by frequency of terms occurring in that time period. The user may also type a new word in the text box at the top of the display and investigate associations with that word. Proceeding in this fashion, the user may explore the Laughlin collection at length and in depth. (We note in passing that a number of other features are supported, but they are a bit peripheral to the central points we wish to make. Given a display as in Figure 3, the user may select a cell and direct the Core of Discovery to display a list of all the underlying documents/records---all 389 records in the case of documents containing "poems" and "louisiana", and associated with photographs taken during the 1940-1944 period. Also, various forms of bar graphs are available for visualizing patterns. These displays carry no more information than is shown in Figure 3, but they may often be more vivid and forceful.)

We now turn to the question of how Homer does all this. After a short discussion of that, we briefly discuss whether Homer is actually effective in helping people find patterns.

How Homer Works

Homer works by displaying results of extensive indexing that is done in batch mode, prior to executing Homer itself. So the key to understanding how Homer works is to understand what the indexing accomplishes. The first step in the indexing process is to divide the document collection (the texts, here the Laughlin comments on the photographs) in a useful, or potentially interesting, fashion. In our current example, see Figure 3, the document collection has been divided into five-year intervals: 1930-1934, 1935-1939, and so on. There happen here to be nine such "bins" into which the documents are categorized. We want to emphasize that there is nothing special about binning in the time domain. Homer can use any categorization available. Moreover, the fact that time is an ordinal variable (1930 is before 1945) is immaterial to Homer. Our bins---our distinct categories in the columns---could just as well have been of a nominal variable, for example state or region in which the photograph was taken. What matters is that at the end of the first step of the indexing the collection of texts is divided into a number of sub-collections. These will correspond to the columns in the Homer display. The binning itself, the division of the documents into the various categories, is done under program control. An indexer, working with a program we call the Core Administrator, indicates how (by what criteria) to divide the document collection and the Core Administrator automatically sets up the records effecting the binning.

The second step in the indexing is to identify all the important words (not including the "and"s and "or"s etc.) in each document in each bin or category. This is done automatically by the Core Administrator program. The result, conceptually, is an indexing array for each category or bin (time period in our present example). This array is often called a term-document matrix (Korfhage, 1997, page 110). We call it K. Rows of the K matrix correspond to indexing terms ("poems", "louisiana", and so on) and columns correspond to documents. (Again: there is one such array for each bin, or time period.) Entries in the array are 1 or 0, depending upon whether the term (corresponding row) occurs in the document (corresponding column). Thus, a small term-document matrix might look like this:

$$K = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The interpretation is that word 1 (whatever that is) occurs in documents 1, 3, 5 and 6, but not 2 or 4. Word 2 occurs in documents 1, 2, 3, 4 and 6, but not 5. And so on. (This K matrix has the same number of rows and columns, but in general that will not be the case.) Creation of a K matrix for each bin/category completes step 2 of the indexing process. Once more: this step is done automatically by the Core Administrator program.

Step 3 completes the indexing process and it, too, is done automatically by the Core Administrator program. Each K matrix is (post-) multiplied by the transpose of itself. We call the resulting matrix L. The L matrix corresponding to the above K matrix is:

$$L = K \bullet K' = \begin{bmatrix} 4 & 3 & 1 & 2 & 1 & 2 \\ 3 & 5 & 2 & 3 & 1 & 2 \\ 1 & 2 & 2 & 2 & 0 & 2 \\ 2 & 3 & 2 & 3 & 0 & 2 \\ 1 & 1 & 0 & 0 & 2 & 1 \\ 2 & 2 & 2 & 2 & 1 & 3 \end{bmatrix}$$

The L matrix always has the same number of rows and columns, this number being equal to the number of rows in the K matrix. Also, L is always symmetric around the northwest-southeast diagonal. But what is important is what L means. Consider the left-most column in our L example. It says that 4 documents (in the bin in question) contain word 1. Of these 4, 3 also contain word 2, 1 word 3, 2 word 4, 1 word 5 and 2 word 6. The interpretation is similar for the other columns/words. That is, word 2 is contained in 5 documents. Of these, 3 contain word 1, 2 contain word 3, 3 word 4, 1 word 5 and 2 word 6.

We are now in position to see exactly what Homer does. Each column in Homer corresponds to an L matrix for its category. By typing in a word---"poems" in our example---the user is choosing a column in the L matrices. Homer displays the "poems" column as a column on the screen. Thus, if the user types in "poems" and "poems" corresponds to column 1,254 of the L matrices, then Homer will display column 1,254 of the L matrix for 1930-1934 under the 30-35 column heading. The story is similar for the other column headings (i.e., column 1,254 of the L matrix for 1935-1939 under the 35-40 column heading). So, an almost entirely automatic process, executed by the Core Administrator program, generates the basic information for Homer.

Does Homer Work?

All this is well and good, but in the end what matters is whether Homer, or any other automated system for supporting pattern-oriented queries, actually helps people answer their questions. It is much too early to provide a definitive answer to this question. We have, however, conducted a number of experimental tests of Homer and gotten uniformly positive results. We report here on a representative study, done with the Laughlin data.

In consultation with the primary Laughlin archivist at The Historic New Orleans Collection, we developed a list of 29 true/false questions about Laughlin's oeuvre. Here are a few of the questions:

- Laughlin is trying to portray the magic of witchcraft in his photography.
- Laughlin used slums as a setting for his poetry photography before 1945.
- Laughlin's poetry photography lacked strong, recurrent themes before 1940.
- Laughlin was more interested in European architecture than in American architecture.

A control group was given no help at all on Laughlin and asked to guess the answers to the 29 true/false questions. They averaged 65% correct. A second control group was given the concept retrieval tool in the Core of Discovery (along with basic instruction). That group averaged 76% of the questions answered correctly, and also on average took nearly 32 minutes to complete the questionnaire. Finally, the Homer group (after being given basic instruction) got 85% of the answers correct and took on average 22.5 minutes to complete the questionnaire. The differences among these numbers are all highly significant ($p=0.02$ or less). More importantly, the absolute sizes of the effects are, we think, quite impressive. Much remains to be learned, but there is good reason for optimism.

Conclusion

Homer, with its use of the L matrices, represents just one way in which automatic indexing may be exploited for purposes of pattern-oriented queries in collections of text. Other methods have been conceived and implemented (e.g., Balachandran et al., 1999). Still others will surely be invented. The distinction, so fruitful here, between record-oriented and pattern-oriented questions is but one way of skinning the question-cat. Other distinctions will surely be made and prove useful. In all of these areas we have much to learn, but the prospects are truly exciting.

References

Balachandran, K., Buzydlowski, J., Dworman, G., Kimbrough, S., Shafer, T. & Vachula, W. (1999). MOTC: An Interactive Aid for Multidimensional Hypothesis Generation. *Journal of Management Information Systems*, forthcoming. Also available in PDF at http://opim.wharton.upenn.edu/~sok/ .

Blair, D.C. (1990). *Language and Representation in Information Retrieval*. Amsterdam: Elsevier.

Codd, E.F., Codd, S.B. & Salley, C.T. (1993). Beyond Decision Support., Computerworld 27(30), July 26.

Dhar, V. & Stein, R. (1997). *Seven Methods for Transforming Corporate Data into Business Intelligence*. Upper Saddle River: Prentice-Hall.

Dworman, G. (1996). Homer: A Pattern Discovery Support System. In M.J. Tauber (Ed.) *ACM SIGCHI Conference on Human Factors in Computing Systems, volume Conference Proceedings Companion* (pp. 305-6). Association for Computing Machinery. Also available at http://opim.wharton.upenn.edu/~dworman/ .

Dworman, G. (1998). Pattern Discovery in Organizational Memory. In V. Jacob & R. Krishnan (Eds.) *Proceedings of the Third Joint International Conference on Information Systems and Technology (CIST).* Also available at http://opim.wharton.upenn.edu/~dworman/ .

Dworman, G.O., Kimbrough, S.O., Kirk, S. & Oliver, J. (1997). On Relevance and Two Aspects of the Organizational Memory Problem, University of Pennsylvania, Department of Operations and Information Management working paper. Also available in PDF at http://opim.wharton.upenn.edu/~sok/ .

Hildebrand, D.K., Laing, J.D. & Rosenthal, H. (1977). *Analysis of Ordinal Data*. Newbury Park: Sage Publications.

Korfhage, R.R. (1997). *Information Storage and Retrieval*. New York: John Wiley & Sons, Inc.

Patch, C. (1994). Tell Me a Story: A System for Thematically Querying a Multi-Media Archive. *Spectra* 22(2), 33-37.

Salton, G. & McGill, M.J. (1983). *Introduction to Modern Information Retrieval*. New York: McGraw-Hill Book Company.

Swanson, D.R. (1988). Migraine and Magnesium: Eleven Neglected Connections. *Perspectives in Biology and Medicine* 31(4), 526-557.

Swanson, D.R. (1993). Intervening in the Life cycles of Scientific Knowledge. *Library Trends* 41(4), 606-631.

Van Rijsbergen, C.J. (1979). *Information Retrieval, second edition*. London: Butterworths.