

On Pattern-Directed Search of Archives and Collections

Garett O. Dworman and Steven O. Kimbrough,

The Wharton School of Business, University of Pennsylvania, 3620 Locust Walk, Suite 1350, Philadelphia, PA 19104-6366. E-mail: dworman@grace.wharton.upenn.edu

Chuck Patch

The Historic New Orleans Collection, 533 Royal Street, New Orleans, LA 70130. E-mail: chuckp@hnoc.org

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.

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 seem 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 expect to resolve it here. Our aim in this article 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 5 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*.¹

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). We often find ourselves 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 5 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.² 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, \dots) 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)?³ 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 rare eventuality can be neglected.)

What to do? How, if the SQL SELECT statement will not work, can we possibly support record-oriented questions with an information system? One thing we can do is to transform or rerepresent the records we do have 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 as to make it easier to get answers to

¹ For the sake of getting to the main points, we pass over the many complexities and subtleties of record-oriented questions. In particular, these questions may be distinguished by how broadly or narrowly they are aimed, whether they ask for records that are similar or different, and so on (cf., Gibson, Kleinberg, & Raghavan, 1998; Kleinberg, 1997; and a detailed discussion by Blair, 1990, 1994).

² There are other kinds of pattern-oriented questions. The work on HITS and related topics is especially noteworthy. It focuses on patterns of reference among Web pages (see Gibson et al., 1998; Kleinberg, 1997). One might also focus on patterns of use, as in the analysis of click-stream data.

³ Here, the numerical coding is only for convenience. Relationships may usefully be studied among ordinal or even nominal variables.

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	...
:	:	:	:	:	:

FIG. 1. (Notional) records pertaining to boating accidents.

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 crosstabulation of underlying records. An elementary example should help make the concepts clearer. [See Fig. 1, which shows in schematic form a series of database records concerning boating accidents (the data are hypothetical but realistic).]

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 crosstabulation of wind and visibility, and the nature of the association is now rather plain.

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 con-

tains 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 crosstabulation generalizes to many dimensions, although using more than five or six 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, Buzydlowski, Dworman, Kimbrough, Shafer, & Vachula, 1999; Codd, Codd, & Salley, 1993; Dhar & Stein, 1997; Hildebrand, Laing, & Rosenthal, 1977). Microsoft Excel’s “pivot tables” are an example of cross tabulation.]

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 response times. 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 mi-

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

FIG. 2. Crosstabulation of wind and visibility information in accident data records.

graines 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)

Since Swanson’s publication in 1988, medical research has found empirical support for this hypothesis, and several research efforts studying the connection have ensued (e.g., Gallai, Sarchielli, Coata, Firenze, Morucci, & Abbritti, 1992). This example, and a few others [mainly from Swanson, but see Gordon & Lindsay’s (1996) replication experiments], 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.

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.

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 Kimbrough & Oliver (1994) and Dworman, Kimbrough, Kirk, & Oliver (1997).

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, 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. To see what Homer does consider Figure 3, which presents Homer’s main (and for our purposes, only) screen.

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 Laughlin himself 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

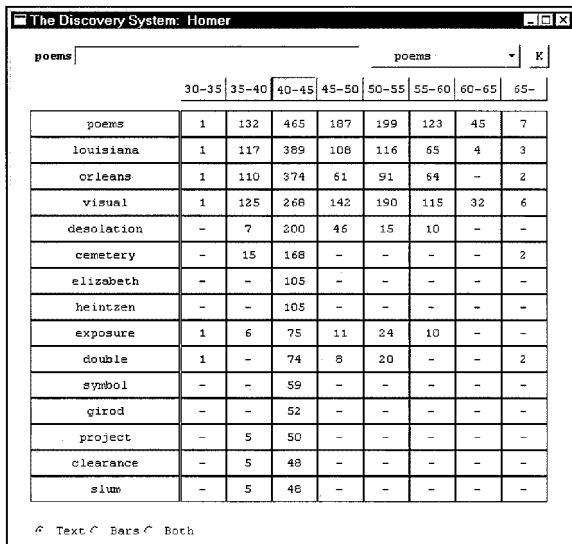


FIG. 3. Homer display.

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.” 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 then produced the display shown in Figure 3.⁴ The first line of the table tells us how “poems” is distributed through the collection: There is one photograph in the 1930–1934 interval with “poems” appearing in Laughlin’s text; 132 photographs in the 1935–1939 interval; 465 in the 1945–1949 interval; and so on. Homer has sorted terms in the collection (excepting “stop” words like “the,” “a,” “to,” . . .) in descending order for the chosen column (1940–1944) by the frequency that they occur within the 465 photographs containing “poems.” Thus, the word “louisiana” occurs in 389 of the 465 photographs within the 1945–1949 interval that contain “poems”; 374 contain “orleans”; and so on down the column. By clicking on a different column heading (corresponding to a different 5-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.

A number of other features are supported. 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,

⁴ A later version of Homer displays the total number of records (photographs) in each time interval, but this version is not currently installed in New Orleans.

various forms of bar graphs are available for visualizing patterns. These displays are often more vivid and forceful than the pure text.

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 Fig. 3), the document collection has been divided into 5-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 categorizing available. Moreover, the fact that time is an ordinal variable (1930 is before 1945) is immaterial for 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 subcollections. 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. The Core Administrator program does this automatically, using a stop list stored in the system and chosen by the user. (We do not currently employ stemming.) 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, p. 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, here time period, for each categorizing variable.) 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 happens to have 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.

The third step creates the cooccurrence data used by Homer. The Core Administrator multiplies each K matrix by its transpose to produce a term-term matrix, L . L is a symmetric matrix with the same number of rows (and columns) as the number of rows in the K matrix. The L matrix corresponding to the above K matrix is:

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

The left-most column in our L example says that four documents (in the bin in question) contain word 1. Of these four, three also contain word 2, one word 3, two word 4, one word 5, and two word 6. The interpretation is similar for the other columns/words. That is, word 2 is contained in five documents. Of these, three contain word 1, two contain word 3, three word 4, one word 5, and two word 6.

The L matrices are the output of the Core Administrator and the input for Homer. Each column in Homer corresponds to an L matrix for its category (or bin of a categorizing variable). When a user enters a term—"poems" in our example—Homer selects the L matrix row for "poems" from the chosen column's L matrix. Homer then sorts and displays the top values from this L matrix row in the selected column, and displays the corresponding terms in the far left column. The rest of the table is filled in by looking up the appropriate values in the other L matrices.

Does Homer Work?

What patterns can we find with Homer? In Figure 3, the following stylistic, temporal, and geographic patterns are evident:

- The most notable topics or shot locations for POEMS in the early 1940s are cemeteries and slums, but neither are used again after 1945 (with the exception of 2 cemetery shots after 1965).
- Nearly all of the POEMS photos before 1960 were taken in New Orleans, LA. However, this location then gradually lost its prominence.
- Stylistically, Laughlin made use of double exposures with his POEMS photos, especially in the early 1940s.
- About 1/4 of the POEMS photos taken in, and only in, the 1940–1945 time frame include the terms ELIZABETH and HEINTZEN. One might surmise that a woman named Elizabeth Heintzen featured prominently in these photos.

To verify that these patterns in the collection correlate to actual stories about Laughlin himself we can look to his own writings. Laughlin grouped his photographs into various categories. For one of his earlier categories, "Poems of the Interior World," he states that "my intent was to present settings for the drama of the misery and madness of our time (Davis, 1990)." Certainly, slum clearance projects and cemeteries seem fitting stages for such a theme. Also, it can be independently determined that Laughlin often traveled from his home in New Orleans later in his life; that double exposures were often employed by Laughlin during his middle period; and, that he Laughlin married Elizabeth Heintzen in 1943.⁵

A critical point here is that none of the above stories about Laughlin were explicitly contained in the collection's documents. The collection did not contain any writing except the photographs' titles and captions that allusively describe the individual photographs' symbolism but not Laughlin's oeuvre as a whole. None of Laughlin's captions discuss his marriage to Elizabeth Heintzen. Nevertheless, Homer, through simple automated indexing, statistical processing, and information presentation, was able to automatically extract enough information to aid the discovery of interesting, nontrivial patterns about Laughlin.

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.

⁵ Personnal communication from John Lawrence, Curator of The Historic New Orleans Collection.

- 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. (See Garrett O. Dworman's forthcoming Ph.D. dissertation at the University of Pennsylvania for details on other experiments.)

Discussion

When Can It Work?

What does it take for a collection or archive to be treated usefully with Homer? Is there anything special about the Laughlin collection, or the other collections that have been examined with Homer, that would prevent generalization to other collections? We propose to answer these questions in two ways: in general, and in particular, with respect to the Laughlin collection. First the general answer.

In pattern-oriented retrieval we are essentially looking at the interplay of two or more categorizing variables. Clearly, then, if pattern-oriented retrieval is to be applied to any collection or archive, it is required that the collection or archive be categorized with at least two variables. This is easily seen for data, as in Figure 2 where the two variables categorizing the collection (of database records) are *wind* and *visibility*. There, and in general, each categorizing variable must be present in at least two forms, or bins: *wind* in Figure 2 has five forms or bins, and *visibility* has three. This much is necessary for any pattern-oriented retrieval method: if you are going to examine the interplay of two or more variables, you have to have them available.

What Homer does, in particular, is to use word (or term) count data, extracted from the collection as one of two possible dimensions (variables). In Figure 3, the *term count* variable is displayed on the Y-axis of the table, much as the *wind* variable is in Figure 2. (Homer supports additional operations, such as sorting; the correspondence is otherwise exact.) *Term count* is always one of the two variables examined by Homer. What the second variable is is application specific. In the case of Figure 3, and the Laughlin collection, the second categorizing variable is *time period* (for the photographs in question). Time is naturally an

interesting second variable, but there is—to repeat—no requirement from Homer to use it. In fact, the user interface for Homer supports an arbitrary number of second variables. The user simply chooses which one to view. Other applications of Homer have used other second variables, including: gender (of a patient), medium (of an artifact), f-stop settings (for photographs), and much more.

As long as the original collection or archive can be partitioned on a given variable, that variable can be used by Homer. What has to be the case for that to happen? Either appropriate data (or metadata) must be present, or the original records must be processed to extract the required data. We have experience with both. In the case of the Laughlin collection, The Historic New Orleans Collection maintains an extensive and detailed database containing information about Laughlin's photographs. (In fact, the texts written by Laughlin and used in the indexing are part of this database.) We used date information from the database to partition the collection and to produce the data displayed in Figure 3. In another of The Historic New Orleans Collection's archives, the Randall Vidrine collection, only text files exist (written by the artist). These had to be processed so that useful variables (and their values)—such as date, exposure, title, and medium (of a photograph)—could be extracted and used by Homer (see <http://www.practicalreasoning.com>).

In sum, any pattern-oriented retrieval system will need to partition its collection of data by two or more variables (categories). Our particular system, Homer, always uses two variables (at a given time), one of which is term count data, which is extracted from the collection using standard methods commonly used in information retrieval systems. Second variables in Homer are collection specific, and depend crucially on what is in the collection and how it is organized. (For further discussion of these topics, as well as description of an innovative exploration tool, see Balachandran et al., 1999.)

Speculations

Having come this far, we find it appropriate to engage in some minor speculations regarding the potential scope of these ideas. Specifically, we ask: When is Homer, or pattern-oriented retrieval in general, likely to be valuable? This important question has at least two senses: (a) When, if we have the necessary categorizing variables, is pattern-oriented retrieval likely to be valuable? And (b) Under what conditions will we likely have the necessary categorizing variables?

Sense (a) first. Pattern-oriented retrieval (and exploration) will be valuable when there are pattern-oriented questions we want to answer. Homer will be valuable when there are two-variable, pattern-oriented questions we want to answer and word count is, or is a surrogate for, one of the variables. It is our belief that such questions are very common indeed, even if they are often not posed because of assumed lack of means to answer them. In the general case, pattern-oriented querying and exploration is appropriate

whenever we seek general rules to summarize information. In the case of Homer, we are typically not interested in the word associations per se; rather, we are interested in what the words represent or suggest, and how these are associated with other variables. Sometimes, as in our Vidrine example, we might be interested in, say, how f-stops vary by date or subject matter. Usually, however, we are likely to be interested in how a theme or subject matter varies with respect to something else, and the word count variable can serve as a suggestive indicator for the theme or subject matter.

Sense (b) *asks* how, in any practical way, can we get the information required to categorize a collection so that the indexing can be created that supports Homer or some other pattern-oriented retrieval system? We have mentioned two ways this has already been done: extraction of information from collections databases, and direct parsing and processing of textual records. With increasing computerization of collections and archives, and with continued technical progress in parsing text, these approaches can only become more useful. There is, moreover, a third approach that promises much for these purposes. We shall now briefly discuss that prospect.

There are excellent reasons why XML (WC3, <http://www.w3.org/XML/>), RDF (WC3, <http://www.w3.org/TR/REC-rdf-syntax/>), and metadata (e.g., Baca, 1998; Bearman, Miller, Rust, Trant, & Weibel, 1999; Lanzi, 1998; Rust, 1998; Dublin Core, <http://purl.org/dc/>; Miller et al., 1999; Weibel, 1999) are “hot topics” and are attracting such impressive amounts of productive attention. Doing commerce electronically—whether one’s purpose is commercial or not—requires increasing structuring of information. Such structuring is necessary for automated processing and support. XML, RDF, and metadata in general are leading banners in the march for such structuring of information.

Our suggestion is that this structuring of information—for example, development of metadata conforming to RDF standards and expressed in XML—will be instrumental in facilitating pattern-oriented retrieval and exploration. Briefly, here is why we think this.

Our essential points can be made without loss of generality by taking a simple view of RDF (see WC3, <http://www.w3.org/TR/REC-rdf-syntax/> for details on RDF). RDF is a logical standard for making metadata statements. It is envisioned, although hardly required, that these statements be expressed in XML. For the present, we can focus on RDF using a simpler syntax. An RDF metadata collection consists of one or more statements having the following form:

{predicate, subject, object}.

An example (section 5 of WC3, <http://www.w3.org/TR/REC-rdf-syntax/>) is

{creator, [<http://www.w3.org/Home/Lassila>], “Ora Lassila”}

which has as its intended interpretation⁶

“Ora Lassila is the creator of the resource <http://www.w3.org/Home/Lassila>”

More complicated arrangements are possible, but this simple structure—{predicate, subject, object}—serves for the points we wish to make: (1) predicates may naturally be viewed as classifying variables for pattern-oriented retrieval and exploration, and (2) subjects and objects may be processed to bin their predicates, thereby creating distinct values for the variables, which values may then be compared in a pattern-oriented discovery system, such as Homer.

We note in passing that it is an obvious extension to use subjects or objects in RDF statements to create classifying variables, and then bin using the other elements in the statements.

An illustration should be helpful in making clear our meaning. Consider the Dublin Core (<http://purl.org/dc/>) as an example metadata standard. The Dublin Core contains 15 “elements,” which, for present purposes we may describe as RDF predicates. From the Dublin Core Web page we get this characterization of the elements:

Element Descriptions

1. *Title*
Label: TITLE
The name given to the resource by the CREATOR or PUBLISHER.
2. *Author or Creator*
Label: CREATOR
The person or organization primarily responsible for creating the intellectual content of the resource. For example, authors in the case of written documents, artists, photographers, or illustrators in the case of visual resources.
3. *Subject and Keywords*
Label: SUBJECT
The topic of the resource. Typically, subject will be expressed as keywords or phrases that describe the subject or content of the resource. The use of controlled vocabularies and formal classification schemas is encouraged.
4. *Description*
Label: DESCRIPTION
A textual description of the content of the resource, including abstracts in the case of document-like objects or content descriptions in the case of visual resources.
5. *Publisher*
Label: PUBLISHER
The entity responsible for making the resource avail-

⁶This is the example as it stands. We note in passing that there is ambiguity here. Event semantics and thematic roles might be used to increase clarity. Under that regime, the verb *creator* would take an *Agent* (here, “Ora Lassila”) and a *Theme* (here, the resource). This would allow a process automatically to know that the resource was created by Ora Lassila, rather than vice versa. [See Kimbrough, 1998–99 for details.]

able in its present form, such as a publishing house, a university department, or a corporate entity.

6. *Other Contributor*

Label: CONTRIBUTOR

A person or organization not specified in a CREATOR element who has made significant intellectual contributions to the resource but whose contribution is secondary to any person or organization specified in a CREATOR element (e.g., editor, transcriber, and illustrator).

7. *Date*

Label: DATE

The date the resource was made available in its present form. Recommended best practice is an 8 digit number in the form YYYY-MM-DD as defined in <http://www.w3.org/TR/NOTE-datetime>, a profile of ISO 8601. In this scheme, the date element 1994-11-05 corresponds to November 5, 1994. Many other schema are possible, but if used, they should be identified in an unambiguous manner.

8. *Resource Type*

Label: TYPE

The category of the resource, such as home page, novel, poem, working paper, technical report, essay, dictionary. For the sake of interoperability, TYPE should be selected from an enumerated list that is under development in the workshop series at the time of publication of this document. See <http://sunsite.berkeley.edu/Metadata/types.html> for current thinking on the application of this element

9. *Format*

Label: FORMAT

The data format of the resource, used to identify the software and possibly hardware that might be needed to display or operate the resource. For the sake of interoperability, FORMAT should be selected from an enumerated list that is under development in the workshop series at the time of publication of this document.

10. *Resource Identifier*

Label: IDENTIFIER

String or number used to uniquely identify the resource. Examples for networked resources include URLs and URNs (when implemented). Other globally unique identifiers, such as International Standard Book Numbers (ISBN) or other formal names would also be candidates for this element in the case of off-line resources.

11. *Source*

Label: SOURCE

A string or number used to uniquely identify the work from which this resource was derived, if applicable. For example, a PDF version of a novel might have a SOURCE element containing an ISBN number for the physical book from which the PDF version was derived.

12. *Language*

Label: LANGUAGE

Language(s) of the intellectual content of the resource. Where practical, the content of this field should coincide with RFC 1766. See: <http://ds.internic.net/rfc1766.txt>

13. *Relation*

Label: RELATION

The relationship of this resource to other resources. The intent of this element is to provide a means to express relationships among resources that have formal relationships to others, but exist as discrete resources themselves. For example, images in a document, chapters in a book, or items in a collection. Formal specification of RELATION is currently under development. Users and developers should understand that use of this element is currently considered to be experimental.

14. *Coverage*

Label: COVERAGE

The spatial and/or temporal characteristics of the resource. Formal specification of COVERAGE is currently under development. Users and developers should understand that use of this element is currently considered to be experimental.

15. *Rights Management*

Label: RIGHTS

A link to a copyright notice, to a rights-management statement, or to a service that would provide information about terms of access to the resource. Formal specification of RIGHTS is currently under development. Users and developers should understand that use of this element is currently considered to be experimental.

Our previous RDF example, concerning Ora Lassila's home page, serves to illustrate RDF representation of element 2, CREATOR.

Notice now that given 15 elements/predicates, there are ${}_{15}C_2 = 105$ possible two-way comparisons that might be explored (and ${}_{15}C_3 = 455$ three-way comparisons!). Many of these would seem to be interesting to someone. How does CREATOR vary by SUBJECT? How does CREATOR vary with FORMAT? How does X (anything) vary by DATE or by PUBLISHER? It would be surprising if an organization went to the trouble of creating metadata descriptions of information and none of the element/predicate associations were interesting to anyone.

Binning is a final consideration here. For objects such as DATE values there are the sort of format issues identified above in the Dublin Core element listing. Assuming these are resolved, it is a straightforward matter for a user to choose intervals (date bins) with appropriate sizes. DESCRIPTION objects typically will be given in free text, and thus present more of a challenge for purposes of binning. Here, binning on word count (after applying stop lists, stemming, and thesauruses)—as in Homer—may be the best we can do. Elements with controlled vocabularies, such as SUBJECT, should be much more easily binned. In short, once variables are identified (e.g., the elements in the Dublin Core) their values must be binned for purposes of analysis. Binning is a nontrivial problem, but it would appear that useful things can be done.

This ends our speculations. We find it plausible that advances in the application of metadata will greatly facilitate pattern-oriented explorations of collections and archives. Demonstrating that is, of course, a matter of science,

not speculation. We hope our remarks here will stimulate such investigations.

Conclusion

Homer, with its use of the *L* matrices, represents one way in which (largely) 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.

Acknowledgments

File: mw99-jasis-19990514.doc. From: mw99-19990128.doc. This is a revised and expanded version of "Pattern-Directed Search of Archives and Collections," which appeared in the conference proceedings of Museums and the Web '99. The authors would like to thank David Bearman for a number of insightful comments and suggestions, and Eric Zheng for several useful references. This material is based upon work supported by, or in part by, DARPA Contract DASW01 97 K 0007.

References

- Baca, M. (Ed.). (1998). *Metadata: Pathways to Digital Information*. Los Angeles, CA: Getty Information Institute. ISBN: 0-89236-533-1.
- 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*, to appear. Also available at <http://opim.wharton.upenn.edu/~sok/>.
- Bearman, D., Miller, E., Rust, G., Trant, J., & Weibel, S. (1999). A common model to support interoperable metadata: Progress report on reconciling metadata requirements from the Dublin Core and INDECS/DOI Communities. *D-Lib Magazine*, ISSN 1082-9873, Volumen 5, Number 1, January 1999. <http://www.dlib.org/dlib/january99/bearman/01bearman.html>.
- Blair, D.C. (1990). *Language and representation in information retrieval*. Amsterdam: Elsevier.
- Blair, D.C. (1994). The challenge of document retrieval: Major issues, and a framework based on search exhaustivity and data base size. Working paper, University of Michigan, Ann Arbor.
- Codd, E.F., Codd, S.B., & Salley, C.T. (1993). Beyond decision support. *Computerworld*, 27(30), July 26.
- Davis, K.F. (Ed.). (1990). *Clarence John Laughlin: Visionary photographer*. Kansas City, MO: Hallmark Cards, Inc.
- Dhar, V., & Stein, R. (1997). *Seven methods for transforming corporate data into business intelligence*. Upper Saddle River: Prentice-Hall.
- Dublin Core. (1999). <http://purl.org/dc/>.
- 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. 305–306). 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., 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/>.
- Gallai, V., Sarchielli, P., Coata, G., Firenze, C., Morucci, P., & Abbritti, G. (1992). Serum and salivary magnesium levels in migraine. Results in a group of juvenile patients. *Headache*, 32(3), 132–135.
- Gibson, D., Kleinberg, J., & Raghavan, P. (1998). Inferring Web communities from link topology. *Proc. 9th ACM Conference on Hypertext and Hypermedia*, Pittsburgh, PA: ACM.
- Gordon, M.D., & Lindsay, R.K. (1996). Toward discovery support systems: A replication, re-examination, and extension of Swanson's work on literature based discovery of a connection between Raynaud's and fish oil. *Journal of the American Society for Information Science*, 47(2), 116–128.
- Hildebrand, D.K., Laing, J.D., & Rosenthal, H. (1977). *Analysis of ordinal data*. Newbury Park: Sage Publications.
- Kimbrough, S.O. (1998–99, Winter). Formal language for business communication: Sketch of a basic theory. *International Journal of Electronic Commerce*, Volume 3, Number 2 (Winter 1998–99), (pp. 23–44). <http://grace.wharton.upenn.edu/~sok/sokpapers/1998-9/ijec-si/dumb.dvi>.
- Kimbrough, S.O., & Oliver, J.R. (1994, December). On relevance and two aspects of the corporate memory problem. In Pradubha De and Carson Woo (Eds), *Proceedings of the Fourth Annual Workshop on Information Technologies and Systems* (pp. 302–311).
- Kleinberg, J.M. (1997). Authoritative sources in a hyperlinked environment. Technical report, Department of Computer Science, Cornell University. <http://www.cs.cornell.edu/home/kleinber/>. An earlier version appeared in *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1998.
- Korfhage, R.R. (1997). *Information storage and retrieval*. New York: John Wiley & Sons, Inc.
- Lanzi, E. (1998). *Introduction to vocabularies: Enhancing access to cultural heritage information*. Los Angeles, CA: Getty Information Institute. ISBN: 0-89236-544-7.
- Miller, P., et al. (1999, March 30). Consortium for the computer interchange of museum information (CIMI) guide to best practice: Dublin Core (DC 1.0 = RFC 2413). http://www.cimi.org/documents/meta_peerreview_announce.html.
- Patch, C. (1994). Tell me a story: A system for thematically querying a multi-media archive. *Spectra*, 22(2), 33–37.
- Rust, G. (1998, July/August). Metadata: The right approach: An integrated model for descriptive and rights metadata in E-commerce. *D-Lib Magazine*, ISSN 1082-9873. <http://www.dlib.org/dlib/july98/rust/07rust.html>.
- 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.
- WC3 (1999, May 11). Extensible markup language. <http://www.w3.org/XML/>.
- WC3 (1999, May 11). Resource description framework (RDF) model and syntax specification: W3C recommendations 22 February 1999. <http://www.w3.org/TR/REC-rdf-syntax/>.
- Weibel, S. (1999, April). The state of the Dublin Core metadata initiative. *D-Lib Magazine*. Volume 5, Number 4, ISSN 1082-9873. <http://www.dlib.org/dlib/april99/04weibel.html>.