Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods

Paru dans *International Journal of Man-Machine Studies*, **38**, 747-767.

ROBERT GODIN, ROKIA MISSAOUI and ALAIN APRIL
Département de Mathématiques et d'Informatique
Université du Québec à Montréal
C.P. 8888, Succursale "A"
Montréal (Québec), Canada, H3C 3P8

ABSTRACT

A controlled experiment was conducted comparing information retrieval using a Galois lattice structure with two more conventional retrieval methods: navigating in a manually built hierarchical classification and Boolean querying with index terms. No significant performance difference was found between Boolean querying and the Galois lattice retrieval method for subject searching with the three measures used for the experiment: user searching time, recall and precision. However, hierarchical classification retrieval did show significantly lower recall compared to the two other methods. This experiment suggests that retrieval using a Galois lattice structure may be an attractive alternative since it combines a good performance for subject searching along with browsing potential.

1. Introduction

Information retrieval is concerned with the representation, storage, organization, and accessing of information items (Salton & McGill, 1983). As opposed to the traditional field of database systems, the items are not restricted to fixed format structured records. Two basic information retrieval methods have gained widespread acceptance: Boolean querying using index terms and navigation in a hierarchical classification. This article presents results of a controlled experiment comparing a new retrieval method based on navigation in a Galois lattice structure with these more conventional methods.

Boolean querying is typical of most online bibliographic retrieval systems. The items are represented by index terms which are used for retrieval. Retrieval is based on direct querying by specification of a Boolean combination of index terms. The matching items are extracted and returned to the user. Using a hierarchical classification is the principal method for organizing and finding files in a computer system. The user simply navigates in the hierarchy by jumping from one class to another using menus or some direct representation of the hierarchy.

The direct querying capacity of Boolean systems is most useful when the user has a good idea of what he is looking for, and can formulate it in the system's language. This is the case for expert intermediaries doing subject retrieval, that is retrieval on a well defined subject. However, query formulation is a difficult process for casual users who may be unfamiliar with Boolean logic, the indexing language and the contents of the database. Many enhancements of the basic approach have been studied such as using relevance feedback, using a thesaurus, extended Boolean retrieval (Salton, 1989).

As pointed out by many researchers (Conklin, 1987; Landauer, Dumais, Gomez & Furnas, 1982; Oddy, 1977; Thompson & Croft, 1989), an information retrieval system should also provide browsing mechanisms for users who do not know precisely what they want or how to get it. Browsing is based on freely exploring a structure such as a tree or a graph in order to find useful items. Browsing in some form of data space has long been recognized as an attractive alternative for information retrieval particularly for casual users and exploration of new domains (Marchionini & Shneiderman, 1988). This form of interaction capitalizes on a well known human cognitive characteristic: it is easier to recognize some object than to describe it.

Navigating in a hierarchical classification is one major browsing paradigm. An early system geared towards interactive browsing in a classification is the ZOG system (Robertson, McCraken & Newell, 1981). The BROWSE system (Palay & Fox, 1981) built on top of ZOG attempted to help the browsing process by providing a partial map of the classification. Today, many modern file systems are based on hierarchical arrangements of folders and documents that can be browsed interactively. Eirund and Kreplin (1988) propose classifying documents and links into IS-A hierarchies which can be browsed. In most systems, the classification structures are maintained manually which is very difficult and time-consuming. Another problem with classification hierarchies is the rigid tree structure where there is only one path from the root to a given class or item. Because of the well known problem of category mismatch (Furnas, Landauer, Gomez & Dumais, 1983), making one bad decision is fatal in traversing the hierarchy and may result in low recall.

Another more recent browsing paradigm is reflected in the current trend of hypertext and hypermedia technologies (Conklin, 1987; Nielsen, 1990; Weyer & Borning, 1985) where the data space is a network of nodes (documents) related by links that may be followed to find related nodes. Typically, the network which supports the browsing has to be manually constructed which is a major bottleneck particularly for large databases. Some possibilities are being investigated for automatically building the network based on some document similarity measure (Thompson & Croft, 1989) or automatic computing of a hierarchy of document clusters for browsing in a restricted set of documents similar to some selected document (Crouch, Crouch & Andreas, 1989). Other browsing tools are also used but are more limited in scope such as browsing a thesaurus for helping the user in the query formulation process. Many recent systems incorporate this possibility (Fowler, Fowler & Wilson, 1991; Frei & Jauslin, 1983; Monarch & Carbonell, 1987).

There have been some empirical work on comparing Boolean querying with navigating in a classification. Geller and Lesk (1983) report on two experiments which compare navigating in a classification using menus and Boolean queries for retrieval with on-line systems. Users could choose between the two alternative methods and their preference was monitored. In a library experiment, users showed clear preference for term searches while the classification browsing method was preferred with a news stories experiment. As suggested, the following reasons may account for some of the differences:

- the users have much more foreknowledge about the library collection as opposed to news which by nature have a dynamic behavior
- browsing retrieval is much more prevalent in news as opposed to library searches where users are often looking for a particular subject or book
- the complexity of the library menus (Dewey Decimal Classification) is much greater than the news menus.

Clearly there are some situations where browsing is preferable and others where direct querying is more satisfactory. What kind of tools should a system provide to the users if all these needs have to be satisfied? User preferences in the previous study suggest that Boolean queries are better for subject retrieval and hierarchical classifications are better for browsing but the study does not evaluate the relative performance of each method.

There is some evidence that index term searching often performs better than classifications for subject searching. A study in (Carpenter, Jones & Oppenheim, 1978) on a small test collection in the field of patent retrieval suggested better performance of index languages over patent classifications. Another small experiment in the same field (Oppenheim, 1980) provided evidence of better recall for index term searching but less precision. A larger experiment by Gerrish (1982) also in the patent retrieval domain compared three index term systems and three classifications on two different types of queries: application-oriented and patentability. Results reveal overall better precision for index term searching but overall recall was better only for the patentability questions. It is important to note that these overall performances were not always reflected for each individual system. Their conclusions should be considered as indicative because of high standard deviations. It should be noted that statistical significance is not addressed in this study and the previous ones. Another problem with these experiments is that the searching is not done by real users using the system but by the people conducting the experiments which may introduce a significant bias. These aspects are taken into account in our experiment.

Most retrieval systems are geared towards one or another of these two types of interactions but it has long been recognized in the context of information retrieval that most searches are a combination of direct and browsing retrieval (Herner, 1970) and as such, a system should provide both possibilities in an integrated and coherent interface. Recognizing the complementary advantages of direct queries and browsing, many hybrid systems offering both modes of interaction have been proposed by combining Boolean queries and hierarchical classifications for example (Frei & Jauslin, 1983; Maarek, Berry & Kaiser, 1991; Marchionini & Shneiderman, 1988; Palay & Fox, 1981; Weyer & Borning, 1985). This approach has some disadvantages. From the systems point of view, both retrieval apparatus have to be maintained since the two components are independent. From the user point of view, only one retrieval method must be used at a time and it is difficult to jump from one method to the other since the two methods are based on independent retrieval spaces.

Using the Galois lattice navigation method may be an interesting alternative since the two modes of interaction may be combined in an integrated and coherent system based on one retrieval space (Godin, Gecsei & Pichet, 1989). This new retrieval method is based on navigating in a particular lattice structure (Godin, Saunders & Gecsei, 1986) also known as a Galois or concept lattice (Barbut & Monjardet, 1970; Wille, 1982; Wille, 1989). The lattice is a network classification structure on the documents that can be automatically generated from the usual document-term indexing relationship. Each class can be thought of as a query described by a subset of terms with the corresponding documents. The network represents the generalization/specialization relationship between classes. Retrieval is performed by a free combination of, (1) direct term specification resulting in a jump to the most general class described by the specified terms and, (2) browsing by following the edges of the network. Following an edge can be interpreted as producing a minimal enlargement (generalization) or refinement (specialization) of the current class. Efficient incremental algorithms have been devised for building the lattice, adding new documents one by one (Godin, Missaoui & Alaoui, 1991). The most attractive feature of the lattice retrieval method is its ability to combine browsing retrieval and direct term queries within a unique retrieval structure.

An important question is the relative retrieval performance of this new method compared to conventional ones. The experiment presented here addresses one performance aspect: subject retrieval. The experiment was performed in a controlled environment where real users had to use three methods (Boolean querying, navigating in a hierarchical classification and navigating in a Galois lattice) for the task of subject searching and three performance measures were considered: user search time, recall and precision. Given the previous experiments and since Boolean queries are more expressive and oriented towards this kind of task, we may anticipate a better performance than for lattice or hierarchical classification retrieval. Also, lattice retrieval seems to have an advantage over hierarchical classification retrieval because there are many paths to a particular category as opposed to a strict hierarchical classification where each category has exactly one parent. Making one bad decision is fatal in traversing a hierarchy but can be compensated in the lattice through other paths. These hypotheses are tested in our study. Although there is a clear advantage for the lattice method and hierarchical classification compared to Boolean retrieval for pure browsing, viz. looking around without any specific goal in mind, this study did not aim at evaluating this aspect which is difficult to measure in such an experiment.

The experiment was performed on a small database (113 documents) extracted from the catalog of films and videos of Canada's ONF ("Office National du Film du Canada"). Each of the 20 subjects performed subject retrieval tasks using each of the three methods and three performance measures were collected: user searching time, recall and precision.

The following section introduces the concept of Galois lattice for a binary relation and explains how retrieval is performed with the prototype user interface used in the experiment. Some details are given about the complexity of the lattice. Section 3 presents the details of the experiment and the results are analyzed.

2. Information retrieval using a Galois lattice

The following is a brief description of the Galois lattice retrieval approach implemented in the prototype system used for the experiments. More details on the structure itself, some possible generalizations, large experimental applications, complexity aspects and algorithms for incrementally updating the structure can be found in (Godin, 1989; Godin et al., 1989; Godin et al., 1991; Godin et al., 1986; Mineau, Gecsei & Godin, 1990).

2.1 GALOIS LATTICE FOR A BINARY RELATION

The starting point for building the Galois lattice is the usual document-term binary indexing relation found in conventional retrieval systems as shown in the example of Figure 1. The documents of this example database are a small subset taken from the ONF database used in the experiment. Each document represents a short animation film and is indexed by a set of terms. For example, document 3 is a film entitled "Taratata" (the film titles are not included in the example) and is described by the set of terms {"ANIMAL", "CHILD", "DOG", "FAIR", "FANTASY", "LOVE", "PARADE"}. No term categories such as year or producer were identified although such a possibility can be integrated (Godin et al., 1986). This simple example will be used to explain the lattice structure and the prototype system used in the experiments. From the document-term relationship a classification is automatically generated. The classification corresponds to the graph (Hasse diagram) of a particular lattice structure also known as a Galois lattice (Barbut & Monjardet, 1970). The lattice for the example is shown in Figure 2. Each class is described by the terms common to every document of the class. The following gives formal details about the structure.

Each element of the lattice is a couple, noted (D,T), composed of a document set $D \subseteq \mathcal{D}$ and a term set $T \subseteq \mathcal{T}$ which are related in a particular way. $\mathcal{D}$ is the set of all documents and $\mathcal{T}$ is the set of all terms in the database. Each couple, (D,T), must be a *complete couple* as defined in the following.

Definition 1. Complete couples.

A couple of sets, (D,T), is *complete* with respect to a document-term relationship if the two following properties are satisfied:

1) D is the set of documents described by at least the terms of T

2) T is the set of terms common to all the documents of D.

This dual relationship represents a Galois connection (Barbut & Monjardet, 1970) between the document sets and term sets. Given a complete couple (D,T), the term set, T, is a set of terms appearing in a conjunctive query which retrieves D. Furthermore it is the *largest* such set and from this point of view, it can be considered as a kind of maximally specific description of the documents in D. The lattice used for retrieval is the set of all complete couples derived from the document-term relationship.

Definition 2. The Galois lattice H*.

The set H* is the set of all complete couples derived from a document-term relationship.

The set H* is the Galois lattice corresponding to the Galois connection induced by the document-term relation (Barbut & Monjardet, 1970). This lattice structure has many applications such as knowledge representation and acquisition (Godin et al., 1991; Missaoui & Godin, 1991; Wille, 1989). H* is a lattice with the following partial order defined from the corresponding order on the term sets:

$$C_1=(D_1,T_1) < C_2=(D_2,T_2) \quad <=> T_1 \subset T_2.$$

Note that the order on the term sets is reversed for the document sets:

$$T_1 \subset T_2 <=> D_2 \subset D_1.$$

When augmenting the number of terms in a conjunctive query (set T), the number of documents satisfying the query (set D) usually decreases and never increases because the query becomes more specific.

The partial order is used to generate the graph in the following way: there is an edge (C_1,C_2) if $C_1<C_2$ and there is no other element C_3 in the lattice such that $C_1<C_3<C_2$. The graph is usually called the Hasse diagram of the lattice. By convention, when drawing a Hasse diagram, the edge direction is from top to bottom as can be seen in Figure 2. Following an edge in the lattice corresponds to a minimal refinement of the query which is relevant to a particular database. In figure 2, for example, the two following couples are linked by an edge:

C_1 = ({1,2,3,6}, {ANIMAL, CHILD, FANTASY}) and

C_2 = ({1,2}, {ANIMAL, CHILD, FANTASY, MAGIC}).

Following this edge means that, in this database, adding the term "MAGIC" in conjunction with the initial query "{ANIMAL, CHILD, FANTASY}" will give the set containing two documents, {1,2}, thus eliminating documents 3 and 6. This refinement is minimal because there is no choice of additional term which would select a set of documents such as {1,2,3} or {1,2,6} containing the set {1,2} and contained in {1,2,3,6}. Conversely, following an edge upwards corresponds to a minimal enlargement. The Hasse diagram displays all such minimal refinements or enlargements.

Showing these possibilities to the user is a particular form of feedback that can be useful for query modification. The importance of feedback in assisting the user has long been recognized as a fundamental principle in the design of good interfaces. Different approaches have been proposed in the context of information retrieval, such as *relevance feedback* (Salton, Fox & Voorhees, 1985), showing *relevant examples* (Oddy, 1977; Williams, 1984) or the *frequency of the index terms* in the retrieved documents (Ingwerson & Wormell, 1986) for query reformulation. In our approach the lattice shows all *minimal refinements* which are pertinent for the query in a particular database with respect to conjunctive queries.

It is important to note that taking all possible document subsets produces an exponential number of nodes. However, when there is a fixed upper bound, say K, on the number of terms for each document, which is usually the case in practical applications, the worst case complexity of the structure is linearly bounded with respect to the number of documents (Godin et al., 1986):

$\|H^*\| \leq 2^K n$ where n is the number of documents.

This upper bound is exponential in K; however, experience with real applications and theoretical results with randomly assigned index terms show that in practice, $\|H^*\|/n$ is fairly stable and much smaller than this upper bound.Table 1 gives the values of $\|H^*\|$, n, $\|H^*\|/n$ and the mean number of index terms assigned to each document, denoted k, for several test applications. The first three applications are described in more details in (Godin et al., 1986).

For example, in a test performed on a database of 3042 technical reports indexed by a mean number of 11.1 terms per report, the mean number of couples divided by the number of documents is 7.7 which is much less than the $2^{11.1}$ upper bound. The last line of Table 1 corresponds to the test database used in the controlled experiment reported here. It will be described in more detail in section 3. In every case, $\|H^*\|/n < k$.

Furthermore, theoretical estimations based on different values of k suggest that the lattice may grow linearly with respect to the number of terms per document. Using a formula derived in (Godin et al., 1986) for the mean value of $\|H^*\|$ with random assignment of index terms within documents, we have computed the theoretical values for different values of k, keeping other parameters fixed. Figure 3 shows $\|H^*\|/n$ versus n for different values of k. Figure 4, giving a linear regression of $\|H^*\|/n$ versus k, with n=200, reveals a growth very close to linearity. Again, we obtain $\|H^*\|/n < k$.

Another important point for practical use of the lattice is that algorithms for incrementally updating the structure by adding new documents one at a time have been devised. Empirical data on the collection of 3042 technical reports show that adding a new document is made in **O**(n) time and disk I/Os (Godin et al., 1991). With the hypothesis of a fixed upper bound for the number of index terms per document, this is also confirmed by a complexity analysis of the algorithm.

2.2 PROTOTYPE USER INTERFACE

In the following, the prototype interface used for the retrieval experiment is briefly described. Design aspects of such an interface can be found in (Godin et al., 1989). On the conceptual level, the user interacts with the system by navigating from vertex to vertex in the lattice, each vertex representing the *current query* of the user. This may be done through direct selection of a neighboring vertex in the graph. Gradual refinement of the query may be accomplished by successively choosing child vertices and gradual enlargement by choosing parent vertices. This mechanism addresses one of the major drawbacks of systems based on Boolean logic where users have difficulties in controlling the amount of output obtained from a query (Borgman, 1986). Another mode of interaction supported is direct specification of a new term refining the current query. Such term specification results in a direct jump to the smallest vertex containing the terms of the current query plus the new term.

The prototype interface is implemented on a standard small screen Macintosh microcomputer. The window, menu, dialog, and text editing interface tools are used for the user interface (Chernicoff, 1985). The lattice is displayed using three windows. Figure 5 is an example screen layout for the lattice in Figure 2. The "Current query" window displays the current query's term list sorted in alphabetical order and the number of documents retrieved. In Figure 5, the current vertex in the graph has the term set {ANIMAL, CHILD, FANTASY}. The window also contains a text box just below the list of terms for directly editing a new term in order to refine the current query. The "More specific queries" window shows the child vertices corresponding to the current vertex. Each vertex is represented by a box containing the number of documents for the query and the list of *additional terms* relative to the current query. More specific queries are always shown in decreasing order of the number of retrieved documents. The "More general queries" window contains the parent vertices.

An important characteristic of the interface is that only direct neighbors in the lattice are displayed. A more sophisticated interface such as a "fisheye view" (Furnas, 1986) showing a larger portion of the lattice might be more effective but would necessitate a larger high-resolution screen and more computing power (Godin et al., 1989).

For navigation purposes, the user can change the current query to an adjacent query (more general or more specific) simply by pointing and clicking on the corresponding box. Figure 6 shows the result of selecting the more specific query containing the additional term "MAGIC" with respect to the current query in the context of figure 5.

Direct term specification is another mode of interaction and is done by filling the editable text box below the list of terms in the "Current query" window. In Figure 7, the user has specified the new term "COW-BOY" in order to refine the current query "{ANIMAL, CHILD, FANTASY}". The "Term" command in the "Search" menu launches the search for the most general vertex containing the new term and the current query resulting in a jump to that new vertex. Figure 8 shows the result of the search.

The user can always come back to any previous query by clicking the corresponding line in the "History of selected queries" window (Figure 9). The two modes of interaction, lattice navigation and direct term specification, may easily be combined in a complementary manner since they are performed in the same retrieval space. The resulting interface is well adapted to different kinds of users and needs. More experienced users knowing the organization of the database and looking for a particular subject can directly specify some combination of terms. On the other hand, casual users without any prior knowledge of the contents of the system or users without any particular subject in mind can freely navigate through the graph without ever specifying any query. Perhaps more importantly, users with mixed needs as is often the case can combine the two modes of dialogue in a systematic manner starting perhaps by some general term specification then browsing from there on, then specifying some other term, etc. For the task of subject searching, the browsing capacity may help compensate for some of the problems encountered with Boolean querying.

The documents of the current query may be consulted using two additional windows. The "Documents of current query" (Figure 9) window shows the list of document numbers and terms for the documents of the current query. The user may consult the contents of any document in this window by clicking on the corresponding line. This has the effect of opening another window exhibiting the contents of the document. For example in Figure 9, document #3 in the "Documents of current query" window has been selected, therefore opening the window entitled "Taratata" giving a short description of the film.

3. Experimental environment and results

SUBJECTS

Twenty subjects were tested in the experiment. The subjects came from various backgrounds but most of them were computer science students and white collar employees with about half the subjects having no prior computer experience. The student subjects were recruited by publicity in our university and were offered money for their participation.

DATABASE

The experiments were performed on a small database extracted from the catalog of films and videos from Canada's ONF ("Office National du Film du Canada"). The database is a collection of 113 short animation film descriptions. For each film, a text document on a MACINTOSH™ microcomputer was created containing the short description in the catalog. The document's title is the film's title.

A manual classification of these documents was created using the MACINTOSH FINDER™. The icons for the documents were all identical and provided no additional retrieval clues. The window position of the icons was based on alphabetic order of the document names and consequently gave no retrieval clues either. Although there is some evidence that spatial positioning may be helpful in small sets, the effect is relatively negligible (Jones & Dumais, 1986). Figure 10 shows the hierarchical classification created for the experiment[1] as well as the number of documents in each class. Also, each document was manually indexed by an average of 6.53 terms. A total of 333 different terms were used. The same indexing relation was used to generate the Boolean and lattice databases. The final lattice contained 325 complete couples with 784 edges and an average of 2.41 parents (or children) per couple. The classification and term assignment were done by one of the authors. Although this may not be detrimental a priori (Furnas

[1] French was the language used in the experiment and therefore the data was translated for the purpose of this article.

et al., 1983), the effect of different classification and indexing strategies is not taken into account in such an experiment.

SOFTWARE

An important consideration in the choice of software is to minimize as much as possible the effect of the user interface. Since the Macintosh user interface paradigm is well standardized, having each representative software from this environment should help in minimizing the interface effect.

For hierarchical classification browsing, the MACINTOSH FINDER™ system was used. This system is based on a hierarchical classification of the documents into folders which may themselves contain documents and folders. A folder may be opened, creating a new window on the screen. Within this window, each folder and document is graphically represented with an icon labeled by a name at a particular physical position. In this experiment, the documents themselves are text files which could be opened and consulted simply by double-clicking on the document's icon. Figure 11 is a sample screen taken from the test database showing three windows related to three opened folders. The folder labeled "ONF" is the root which contains the seven folders corresponding to the seven subclasses of the root in Figure 10. The "Fantasy" folder appearing in the root folder has also been opened. It contains four folders and three documents. Finally, the "Celebration" folder has also been opened and it contains four documents.

The commercial software FACTFINDER™ was used for Boolean queries. Users formulate a query using standard Boolean logic for combining index terms. Figure 12 is a sample screen showing an example Boolean query, "ANIMAL AND FANTASY AND CHILD", formulated in the text box of the "Find in..." window. To assist the query formulation process, the user can ask for a vocabulary window which permits browsing in the alphabetic list of terms used in all documents. After launching a query, the system responds by giving the number of documents found, called "Factsheets", and the list of extracted document titles ("Names") in a scrolling area appearing in a window entitled "Names Found".

In Figure 12, four documents have been extracted by the query. The contents of a document is displayed in a window by selecting the document title in the list of extracted documents. The document entitled "Taratata" has been selected in Figure 12. The "Keys for..." window displays the list of terms assigned to the document.

EXPERIMENTAL PROTOCOL

Each subject had to use the three methods for the task of subject searching. In order to minimize a possible learning effect, the order of the methods was randomized. For each method, the experiment was preceded by a tutorial session done on a small training database. The tutorial was pursued until the user could successfully perform test subject searches.

For each method, after the tutorial session, each subject had to perform five subject searches randomly selected among a set of 30 queries for which relevance judgments were performed in advance. The choice of queries is a difficult matter and there is no objective way to determine what is a good representative set of queries. However, much care was taken in the wording of the queries in order to be as neutral as possible with respect to the methods. We tried to have a mix of different types of queries: some were formulated with the index terms of the documents, others with the words in the class labels of the classification, some with none of these, some were more general, others very specific, some suggested disjunctive queries, etc. The average number of relevant documents for the 30 queries was 1.83. For each search the user was asked to find documents about a specific subject (e.g. "social impact of new technologies"). For each method, the users could interactively consult the text descriptions on the screen before effectively retrieving a document. They were not pressured to finish within a certain time period. After the search, the documents retrieved were noted but no feedback on the validity of the results was given to the user. For each search, three measures, known to be among the most important from the users point of view (Salton & McGill, 1983), were considered.

1) Search time: the time taken by the user to perform his task. This is a relatively good and objective measure of the effort required from the users in performing the task. Because of the small size of the database, response time was not a detrimental factor which might have introduced a bias.

2) Precision: this is the number of documents retrieved and relevant divided by the number of retrieved documents. Relevance judgments were made in advance for each query. This measures the capacity of a system to give only relevant items.

3) Recall: this is the number of documents retrieved and relevant divided by the number of relevant documents. This measures the capacity of a system to give all relevant items.

The last two measures are the most widely used in laboratory experiments. However, they do not give any measure of the user's effort which is important from the user point of view. The experiment lasted approximately 2-3 hours for each subject.

RESULTS

Table 2 shows the average values and standard deviations for each method. Overall, the Boolean method has the best scores for every measure. The lattice method is second for recall and precision and last for search time. The hierarchical method is last for precision and recall but second for search time. A one factor analysis of variance (ANOVA) with repeated measures (same subjects for each method) was performed for each measure, the method being the fixed factor. Results of the F tests for effect of the method appear in Table 3 .

The ANOVA in Table 3 reveals no effect of the method on search time ($P = 0.60$) although the average values in Table 2-a show that the Boolean method (3.55) performs slightly better than the two other methods which are very close (3.90 for hierarchic and 3.95 for lattice). Also, the method was not significant for precision ($P = 0.70$). This result may be explained by the fact that, in each method, the user could examine the film descriptions before extracting them. This is different from laboratory experiments where the precision is calculated from the documents retrieved by a Boolean query without any relevance judgment on the part of a user. Consequently, in this type of experiment, precision may measure the users judgment more than the effect of the retrieval method. Therefore, we may expect that the users judgment does not change from method to method.

With a value of P = .05 for recall, we can conclude that the effect of the method on recall was significant. Recall for Boolean is slightly better than for the lattice method, 80.9% and 79.5% respectively, but testing for superiority of Boolean over lattice (paired t-test, 1-tail) yields a P value of .37 which is far from being significant (Table 4).

This may seem surprising because, compared to Boolean queries, the expressive power of the lattice is limited, since it only represents the equivalent of conjunctive queries. There is some evidence that the differences may be small since many users do not exploit the full power of expressiveness offered by Boolean queries. Borgman (Borgman & Meadow, 1985) reports that typically, users of on-line systems use conjunctive queries and refine them by adding or deleting terms in the conjunction. Informal observations made during our experiments confirm those of Borgman. This kind of step by step refinement is exactly what the lattice browsing permits and furthermore, the lattice helps the user by suggesting some good refinements with respect to the database under consideration. This additional feedback may compensate for the inferior expressiveness of the lattice. Observations also revealed that, for the lattice method, most users start with a general term specification (direct query) jumping into a node of the lattice and go on browsing from that point. The fact that the database is small may also play in favor of the lattice method. In a large database, the finer discriminating possibilities of Boolean querying may become more apparent.

Recall for the hierarchical method was the worst of the three methods with a value of 70.5%. Testing for superiority of mean recall values (Table 4) for the Boolean method (P = .01) and the lattice method (P = .04) with respect to the hierarchical method, using a paired t-Test, reveals a significant difference in both cases. The lower performance of the hierarchical method compared to Boolean queries might be expected since the user can easily miss the browsing path intended by the people doing the classification because there is only one path from the root down to the pertinent documents. The previously mentioned category mismatch problem for hierarchical classifications is well documented. The browsing environment may not always compensate for the strict structure of the classification. Conversely, with terms, there are many "paths" to reach the documents, which may compensate for language and conceptual ambiguities. The lattice also has this characteristic since it is based on the same terms and there may be many parents for a given class.

An important caution should be given at this point. Even though a uniform user interface environment was chosen, a difficulty in such experiments is that we may still be comparing interfaces more than the underlying methods. The lattice interface in particular was implemented as a prototype with limited time (2 person/month) and resources as opposed to the two others which are more refined commercial products. For example, manipulating the document windows has proven very cumbersome in the prototype lattice software and many subjects showed frustration in this process. There are six windows which may appear simultaneously on the screen and the user has to manage their existence, position and size manually. For example, it would of been useful to automatically adjust the size of the "More general queries" and "More specific queries" windows with respect to the number of parents and children. Also, if the user wants to consult a document, he has to go in the "Documents of current query" window, click in the corresponding line which displays a new window in the center of the screen showing the content of the document, and then close it to continue the querying because the window overlaps the other windows. The equivalent mechanisms used in the Boolean system are more polished: by default the first document of the extracted is always displayed and it does not interfere with the querying window. A similar strategy could of been adopted for the lattice interface.

4. Conclusion

This article presented results of a controlled experiment comparing a new retrieval method based on navigation in a Galois lattice with more conventional methods: browsing in a hierarchical classification and Boolean queries. The experiment was performed by real users using real systems for the task of subject searching. Three performance measures were considered: user searching time, recall and precision.

Boolean querying produced the best performance for every performance measure. However, the performance was far from being significantly better than Galois lattice retrieval. A significant difference was observed for recall using the hierarchical classification retrieval which was lower than the other two methods. It should be emphasized that the users were allowed to see the documents and make relevance judgements before retrieving them. Consequently, precision may measure more the users ability in making relevance judgements than the methods effect. This may explain the fact that the retrieval method did not show any significant effect on precision.

We should also emphasize the fact that such experiments should be interpreted with extreme caution. How will these results transpose to larger databases, different domains, different interfaces, different query sets, different classifications and indexing, is very speculative. Experiments reported by Blair and Maron (1985), for example, revealed in a striking manner that the good performance results reported earlier for full-text retrieval on small collections dropped dramatically for a large scale database. Although we recognize the limits of our experiment, they may give some insight on the value of using a Galois lattice as an alternative approach to information retrieval. Since the lattice method revealed performance comparable to Boolean searching for subject retrieval and has an additional advantage for browsing purposes, it is an interesting alternative to be considered especially for small to medium databases where the cost of updating the lattice structure is less detrimental. Further experiments should aim at determining how the performance results change when controlling different factors such as:

database volume, indexing strategies, classification strategies, interface details, domain of the database, user profile, query characteristics. Also some experiments should be devised to measure concretely the performance of these methods for browsing retrieval.

Acknowledgments

We would like to thank the "Service de Consultation en Analyse de Données" of " l'Université du Québec à Montréal" for their help in planning and analyzing the experiment. We would also like to thank the anonymous referees for their comments which greatly helped in improving the presentation of this paper. This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants No OGP0041899, OGP0009184, EQP0092688 and by FCAR Funds ("Fonds pour la Formation de Chercheurs et l'Aide à la Recherche") under grant No 91-NC-0446.

References

Barbut, M. & Monjardet, B. (1970). *Ordre et Classification. Algèbre et Combinatoire, Tome II*. Hachette.

Blair, D. C. & Maron, M. E. (1985). An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System. *Communications of the ACM*, **28**(3), 289-299.

Borgman, C. L. (1986). Why are Online Catalogs Hard to Use? Lessons Learned from Information-retrieval Studies. *Journal of the American Society for Information Science*, **37**(6), 387-400.

Borgman, C. L. & Meadow, C. T. (1985). Designing an Information Retrieval Interface Based on User Characteristics. In *Proceedings of the Eighth Annual International ACM SIGIR Conference*, pp. 139-146.

Carpenter, A. M., Jones, M. G. & Oppenheim, C. (1978). Retrieval Tests on Five Classification Schemes. *International Classification*, **5**(2), 73-80.

Chernicoff, S. (1985). *Macintosh Revealed: Unlocking the Toolbox*. New Jersey: Hayden.

Conklin, J. (1987). Hypertext: An Introduction and Survey. *IEEE Computer*, **20**(9), 17-41.

Crouch, D. B., Crouch, C. J. & Andreas, G. (1989). The Use of Cluster Hierarchies in Hypertext Information Retrieval. In *Proceedings of the Hypertext'89*, ACM,

Eirund, H. & Kreplin, K. (1988). Knowledge Based Document Classification Supporting Integrated Document Handling. In *Proceedings of the ACM Conference on Office Information Systems*, Palo Alto:

Fowler, R. H., Fowler, W. A. L. & Wilson, B. A. (1991). Integrating Query, Thesaurus, and Documents through a Common Visual Representation. In *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, A. Bookstein,Y. Chiaramella,G. Salton, & V. V. Raghavan (Ed.), Chicago, Illinois: ACM, pp. 142-151.

Frei, H. P. & Jauslin, J.-F. (1983). Graphical Presentation of Information and Services: A User Oriented Interface. *Information Technology: Research and Development*, **2**(1), 23-42.

Furnas, G. W. (1986). Generalized Fisheye Views. In *Proceedings of the Human Factors in Computing Systems (CHI'86)*, pp. 16-23.

Furnas, G. W., Landauer, T. K., Gomez, L. M. & Dumais, S. T. (1983). Statistical Semantics: Analysis of the Potential Performance Of Key-Word Information Systems. *The Bell System Technical Journal*, **62**, 1753-1806.

Geller, V. J. & Lesk, M. E. (1983). User Interfaces to Information Systems: Choices vs. Commands. In *Proceedings of the 6th Annual International ACM Sigir Conference on Research and Development in Information Retrieval,* ACM Sigir Forum, pp. 130-135.

Gerrish, S. H. (1982). *Classifications versus Keywords for Patent Information Retrieval.* London, City University, Center for Information Science.

Godin, R. (1989). Complexité de Structures de Treillis. *Annales des Sciences Mathématiques du Québec*, **13**(1), 19-38.

Godin, R., Gecsei, J. & Pichet, C. (1989). Design of a Browsing Interface for Information Retrieval. In *Proceedings of the 12th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, N. J. Belkin & C. J. van Rijsbergen (Ed.), Cambridge, MA: ACM SIGIR Forum, pp. 32-39.

Godin, R., Missaoui, R. & Alaoui, H. (1991). Learning Algorithms Using a Galois Lattice Structure. In *Proceedings of the Third International Conference on Tools for Artificial Intelligence*, San Jose, Calif.: IEEE Computer Society Press, pp. 22-29.

Godin, R., Saunders, E. & Gecsei, J. (1986). Lattice Model of Browsable Data Spaces. *Information Sciences*, **40**, 89-116.

Herner, S. (1970). Browsing. In M. Dekker (Eds.), *Encyclopedia of Library and Information Science,* (pp. 408-416).

Ingwerson, P. & Wormell, I. (1986). Improved Subject Access, Browsing and Scanning Mechanisms in Modern Online IR. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval*, pp. 68-76.

Jones, W. P. & Dumais, S. T. (1986). The Spatial Metaphor for User Interfaces: Experimental Tests of Reference by Location versus Name. *ACM Transactions on Office Information Systems*, **4**(1), 42-63.

Landauer, T. K., Dumais, S. T., Gomez, L. M. & Furnas, G. W. (1982). Human Factors in Data Access. *The Bell System Technical Journal*, **61**, 2487-2509.

Maarek, Y. S., Berry, D. M. & Kaiser, G. E. (1991). An Information Retrieval Approach for Automatically Constructing Software Libraries. *IEEE Transactions on Software Engineering,* **17**(8).

Marchionini, G. & Shneiderman, B. (1988). Finding Facts vs. Browsing Knowledge in Hypertext Systems. *IEEE Computer*, **21**(1), 70-80.

Mineau, G., Gecsei, J. & Godin, R. (1990). Structuring Knowledge Bases using Automatic Learning. In *Proceedings of the IEEE Sixth Int'l Conf. on Data Engineering*, Los Angeles, CA: IEEE Computer Society Press, pp. 274-280.

Missaoui, R. & Godin, R. (1991). An Expert System for Discovering and Using the Semantics of Databases. In *Proceedings of the World Congress on Expert Systems*, J. Liebowitz (Ed.), Orlando, Florida: Pergamon Press, pp. 1732-1740.

Monarch, I. & Carbonell, J. (1987). CoalSORT: a Knowledge-Based Interface. *IEEE Expert*, **2**, 39-53.

Nielsen, J. (1990). *Hypertext & Hypermedia*. San Diego, California: Academic Press.

Oddy, R. N. (1977). Information Retrieval through Man-Machine Dialogue. *Journal of Documentation,* **33**(1), 1-14.

Oppenheim, C. (1980). The Patent Office Classification Scheme. *The Inventor,* **20**(2), 15-17.

Palay, A. J. & Fox, M. S. (1981). Browsing through Databases. In O. e. al. (Eds.), *Information Retrieval Research,* (pp. 311-324). London: Butterworths.

Robertson, G., McCraken, D. & Newell, A. (1981). The ZOG Approach to Man-Machine Communication. *International Journal of Man-Machine Studies*, **14**, 461-488.

Salton, G. (1989). *Automatic Text Processing*. Reading, Massachusetts: Addison-Wesley.

Salton, G., Fox, E. A. & Voorhees, E. (1985). Advanced Feedback Methods in Information Retrieval. *Journal of the American Society for Information Science*, **36**, 200-210.

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

Thompson, R. H. & Croft, B. (1989). Support for Browsing in an Intelligent Text Retrieval System. *International Journal of Man-Machine Studies*, **30**, 639-668.

Weyer, S. A. & Borning, A. H. (1985). A Prototype Electronic Encyclopedia. *ACM Transactions on Office Information Systems*, **3**, 63-88.

Wille, R. (1982). Restructuring Lattice Theory: an Approach Based on Hierarchies of Concepts. In I. Rival (Eds.), *Ordered Sets*, (pp. 445-470). Dordrecht-Boston: Reidel.

Wille, R. (1989). Knowledge Acquisition by Methods of Formal Concept Analysis. In E. Diday (Eds.), *Data Analysis, Learning Symbolic and Numeric Knowledge*, (pp. 365-380). New York: Nova Science Pub.

Williams, M. D. (1984). What Makes RABBIT Run. *International Journal of Man-Machine Studies*, **21**, 333-352.

Table 1. Lattice complexity in experimental applications.

Experiment	||H*||	n	||H*||/n	k
Technical reports	23,471	3042	7.7	11.1
Videotex database	450	317	1.4	4.7
Course catalog	237	106	2.2	4.8
ONF animation films and videos	325	113	2.9	6.5

TABLE 2-a. Average values for search time, precision and recall.

Method	Search time (min.)	Precision	Recall
Hierarchical	3.90	83.4%	70.5%
Boolean	3.55	83.5%	80.9%
Lattice	3.95	80.5%	79.5%

TABLE 2-b. Standard deviation for search time, precision and recall.

Method	Search time (min.)	Precision	Recall
Hierarchical	1.55	11.6	19.6
Boolean	1.88	14.5	10.9
Lattice	1.61	18.2	13.8

TABLE 3. Effect of the method for a one factor ANOVA with repeated measures.

Effect of method	Search time	Precision	Recall
F(2,59)	.36	.52	3.17
P value	.60	.70	.05

TABLE 4. Comparing mean recall values using a paired t-Test.

Paired t-Test	Boolean > lattice	Boolean > hierarchic	lattice > hierarchic
t	.317	2.46	1.83
P (1-tail)	.37	.01	.04

Document number	**Index terms**
1	{ANIMAL, BEAR, CANADA, CHILD, COW-BOY, DREAM, FANTASY, IMMIGRATION, INDIAN, MAGIC}
2	{ANIMAL, CAT, CHILD, FANTASY, MAGIC, TALE}
3	{ANIMAL, CHILD, DOG, FAIR, FANTASY, LOVE, PARADE}
4	{CHILD, FANTASY, FRIENDSHIP, GAME, ROPE}
5	{CREATIVITY, CHILD, FANTASY, GAME, MUSIC, SOUND}
6	{ANIMAL, CHILD, DREAM, FAIRY, FANTASY, FRIENDSHIP, OCTOPUS}

Figure 1. Example indexing relationship for a subset of the ONF database.

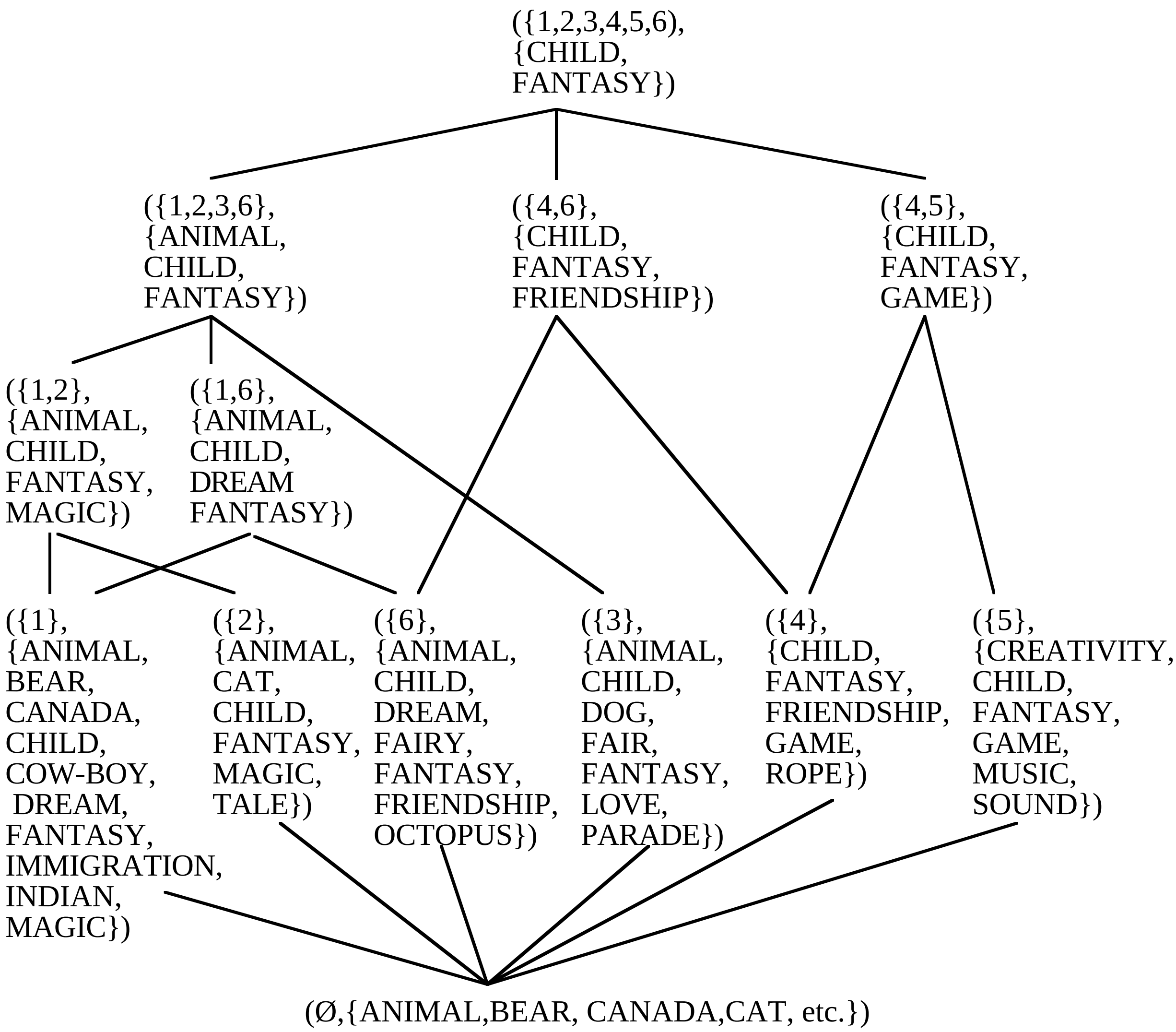


Figure 2. Graph of the Galois lattice for the example indexing relationship of figure 1.

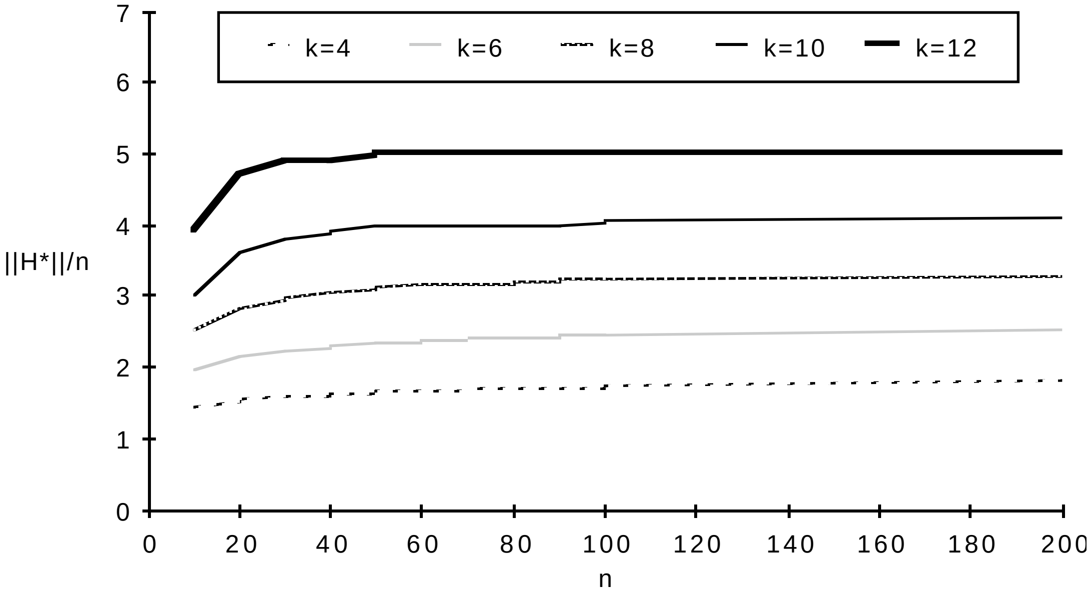


Figure 3. Theoretical values of ||H*||/n versus n for different values of k.

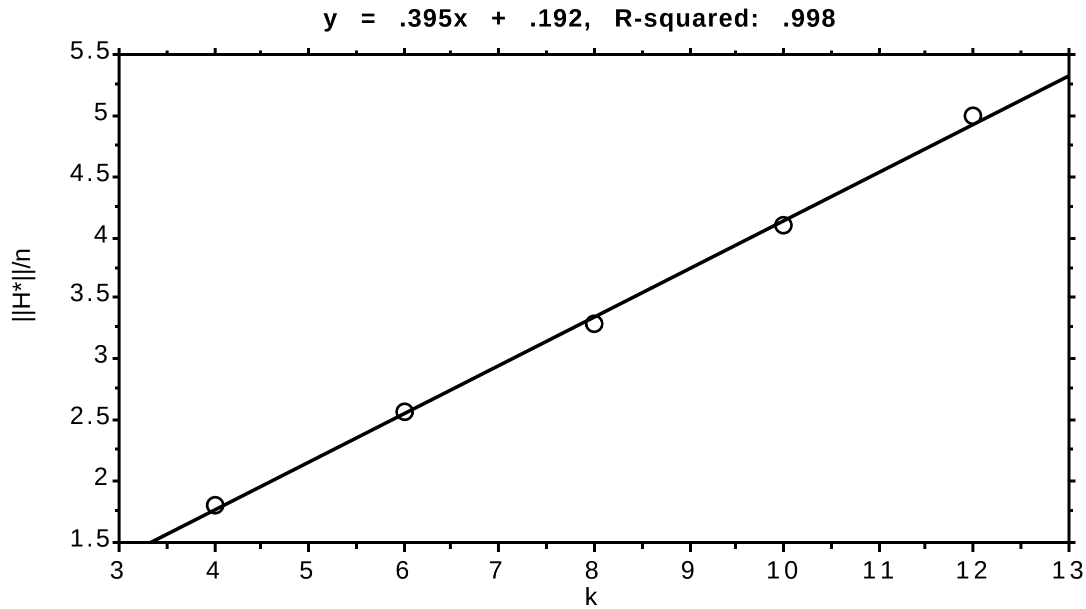


Figure 4. Regression of ||H*||/n versus k for n = 200.

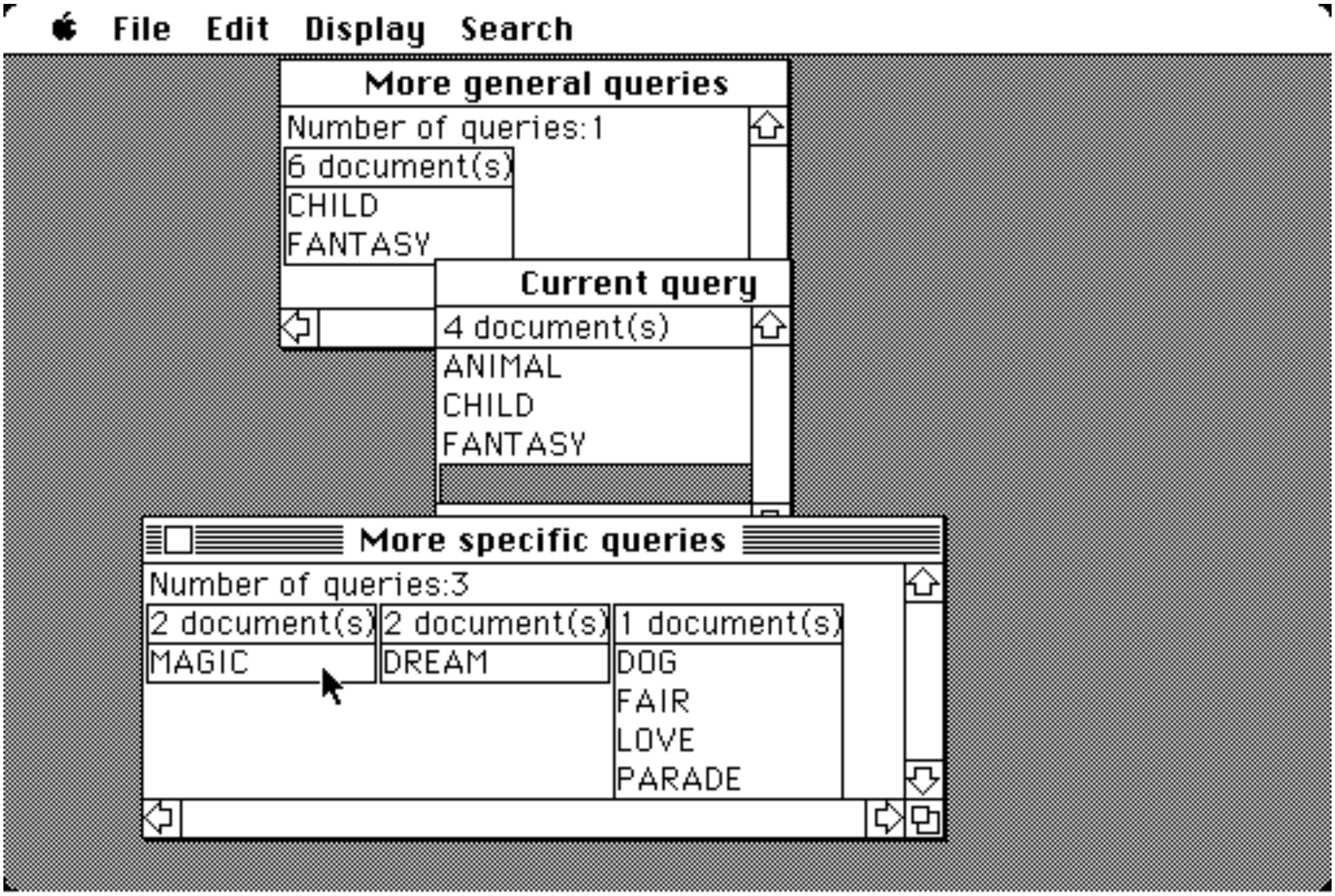


Figure 5. Example screen layout for the lattice in Figure 2. The cursor is positioned for the selection of the more specific query which adds the term "MAGIC" to the current query.

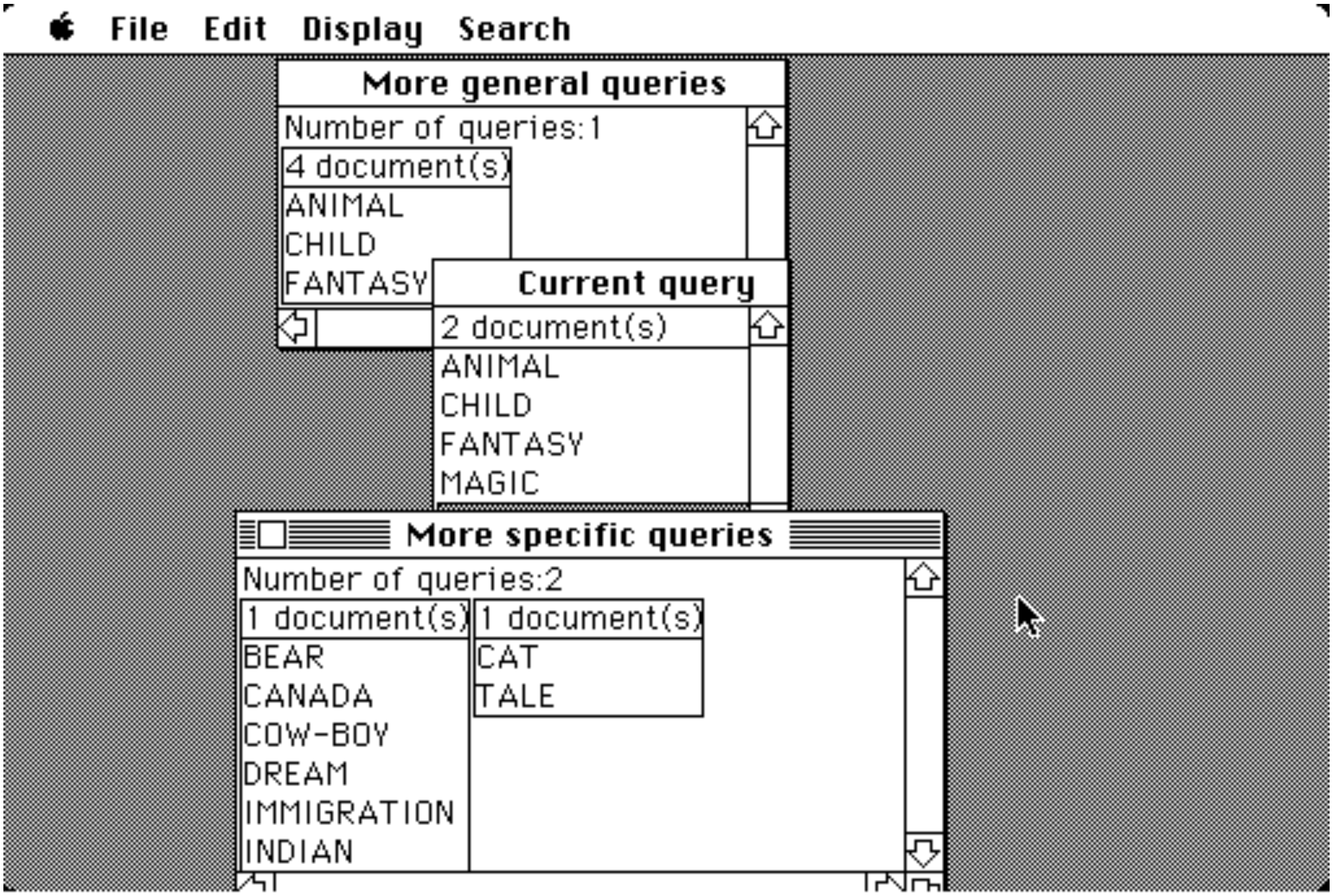


Figure 6. Result of the selection of figure 5.

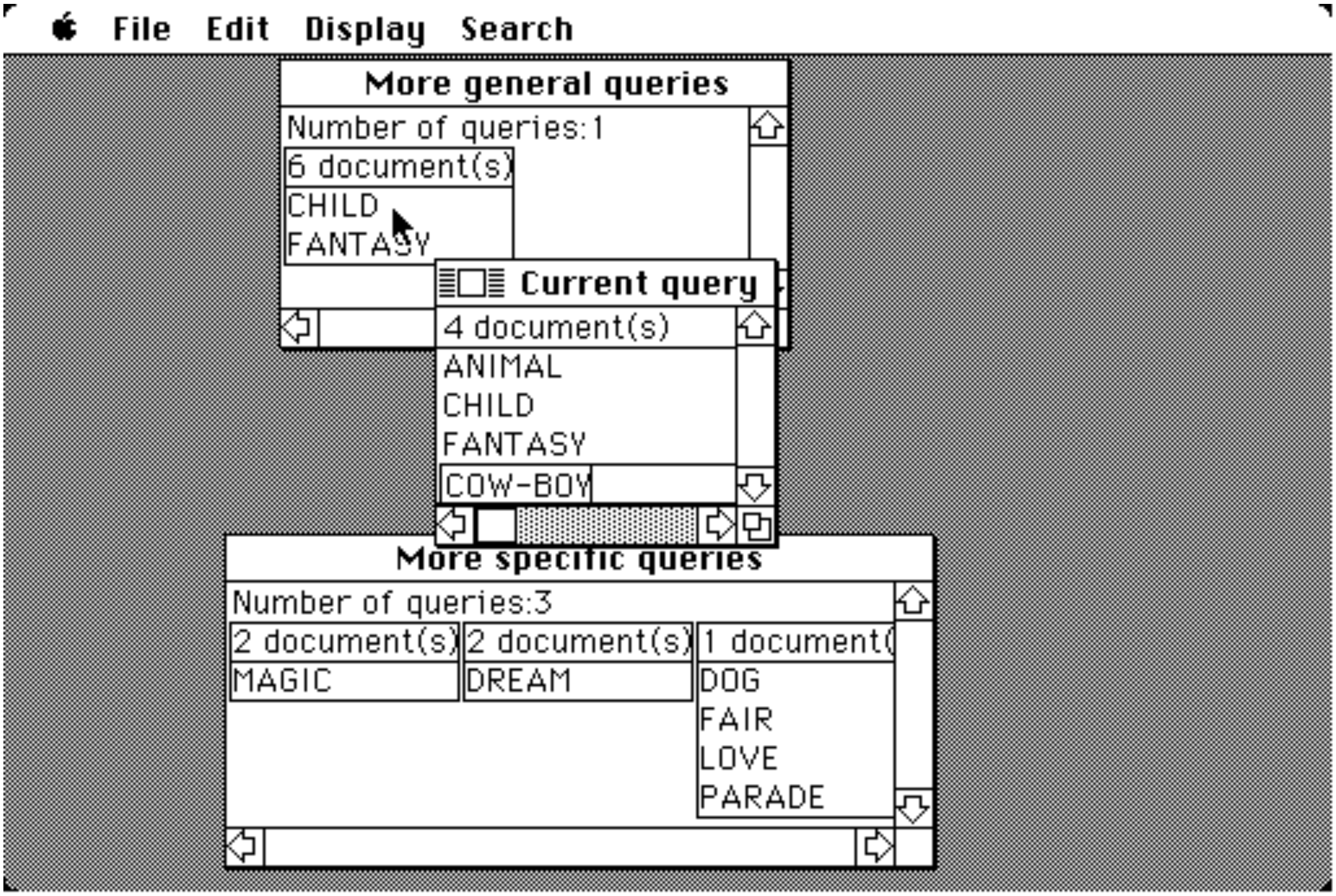


Figure 7. Specification of the new term "COW-BOY" editing it in the text bar of the "Current query" window. The search is launched by selecting "Keyword" in the "Search" menu or by typing <Enter>.

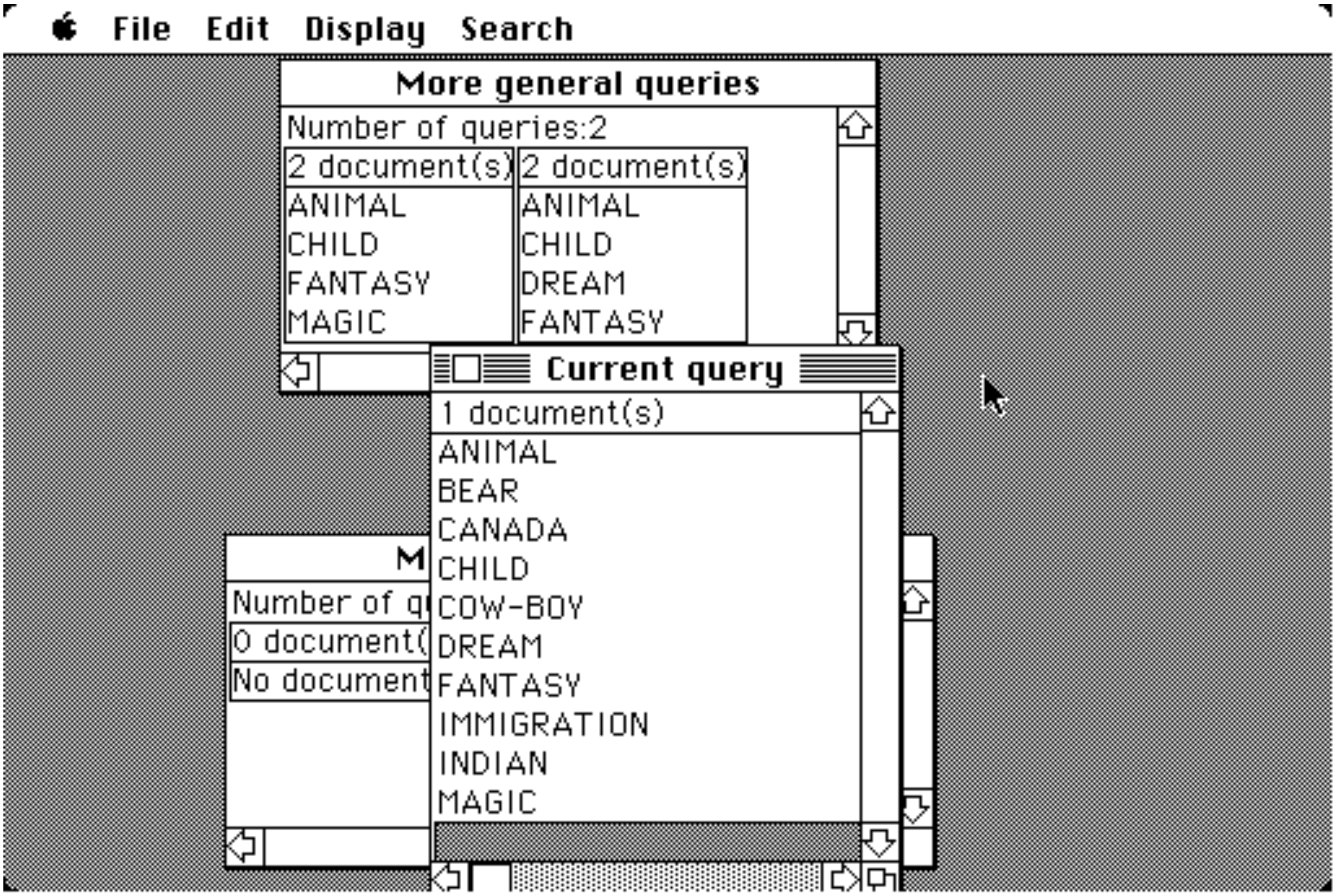


Figure 8. Result of searching the term "COW-BOY" in the context of the preceding current query "ANIMAL, CHILD, FANTASY". The most general query in the lattice containing both the terms "ANIMAL, CHILD, FANTASY" and "COW-BOY" becomes the current query.

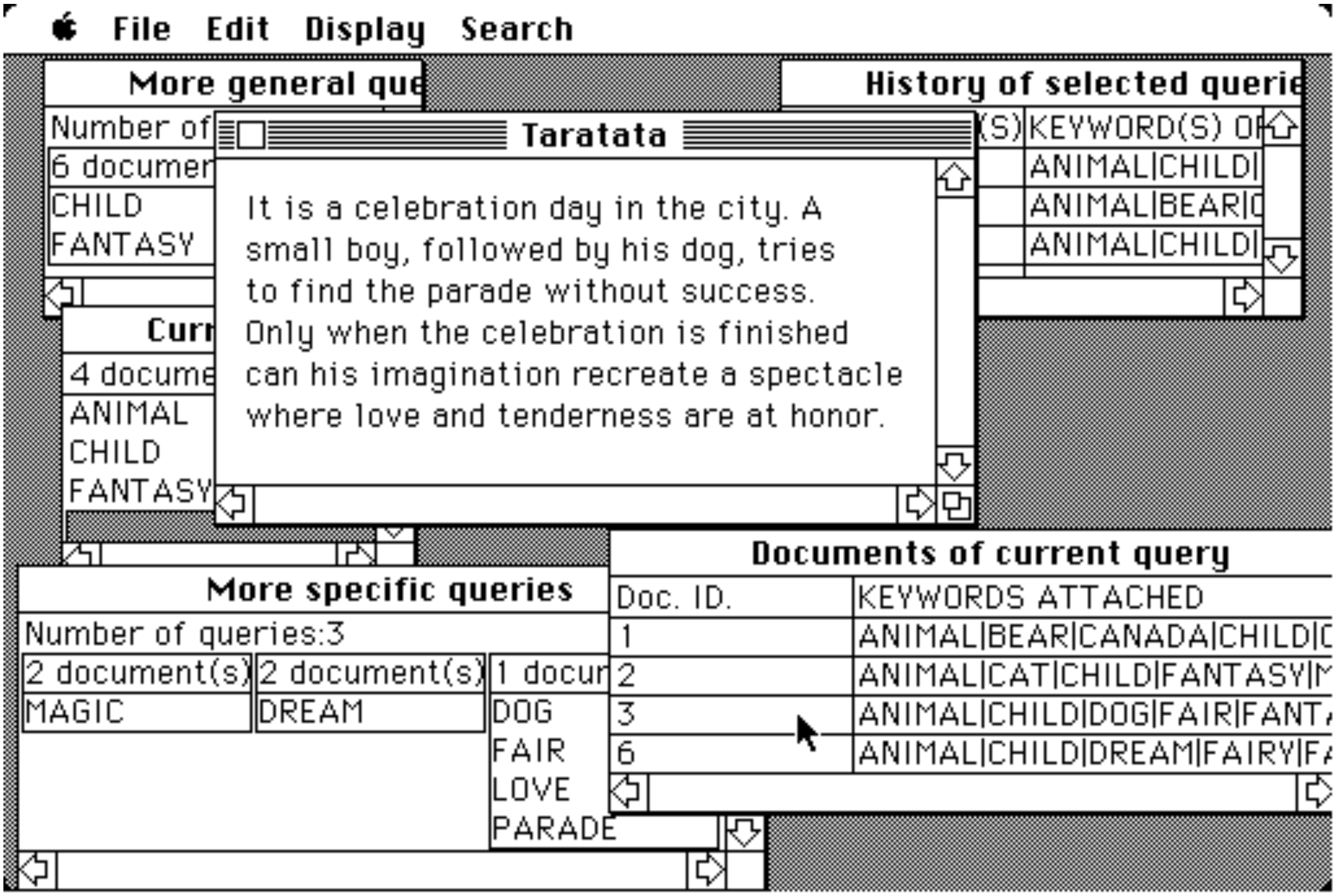
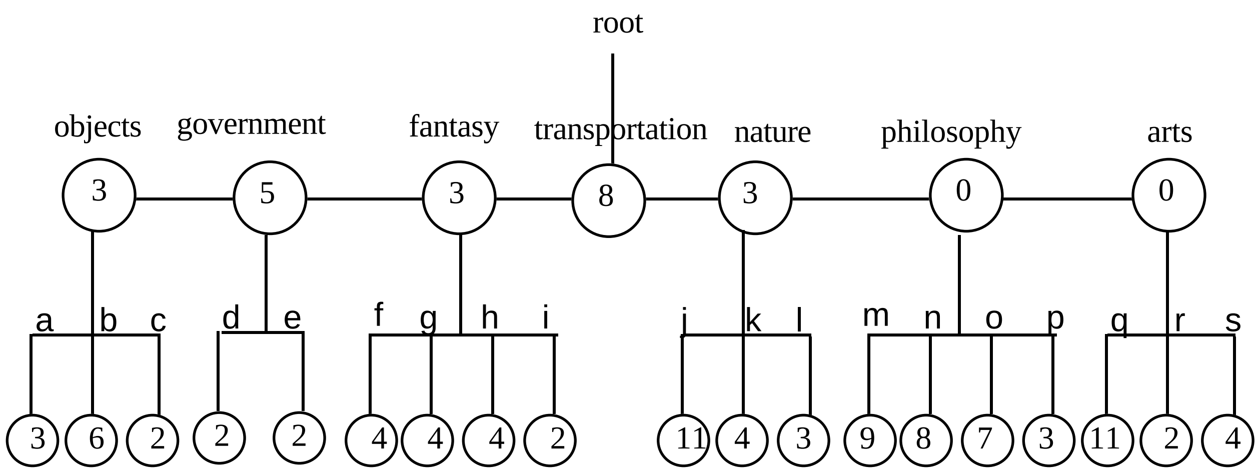


Figure 9. Example screen showing the "History of selected queries", the "Documents of current query" and a document window entitled "Taratata" corresponding to the selection of the document #3 of "Documents of current query" window.



a: chair b: geometry c: color d:taxes e: propaganda
f: treasure g: love h: celebration i:dream j: animal
k: pollution l: planet m: human n:society o:life
p: technology q:song r:music s:dance

Figure 10. Hierarchical classification for the ONF database.

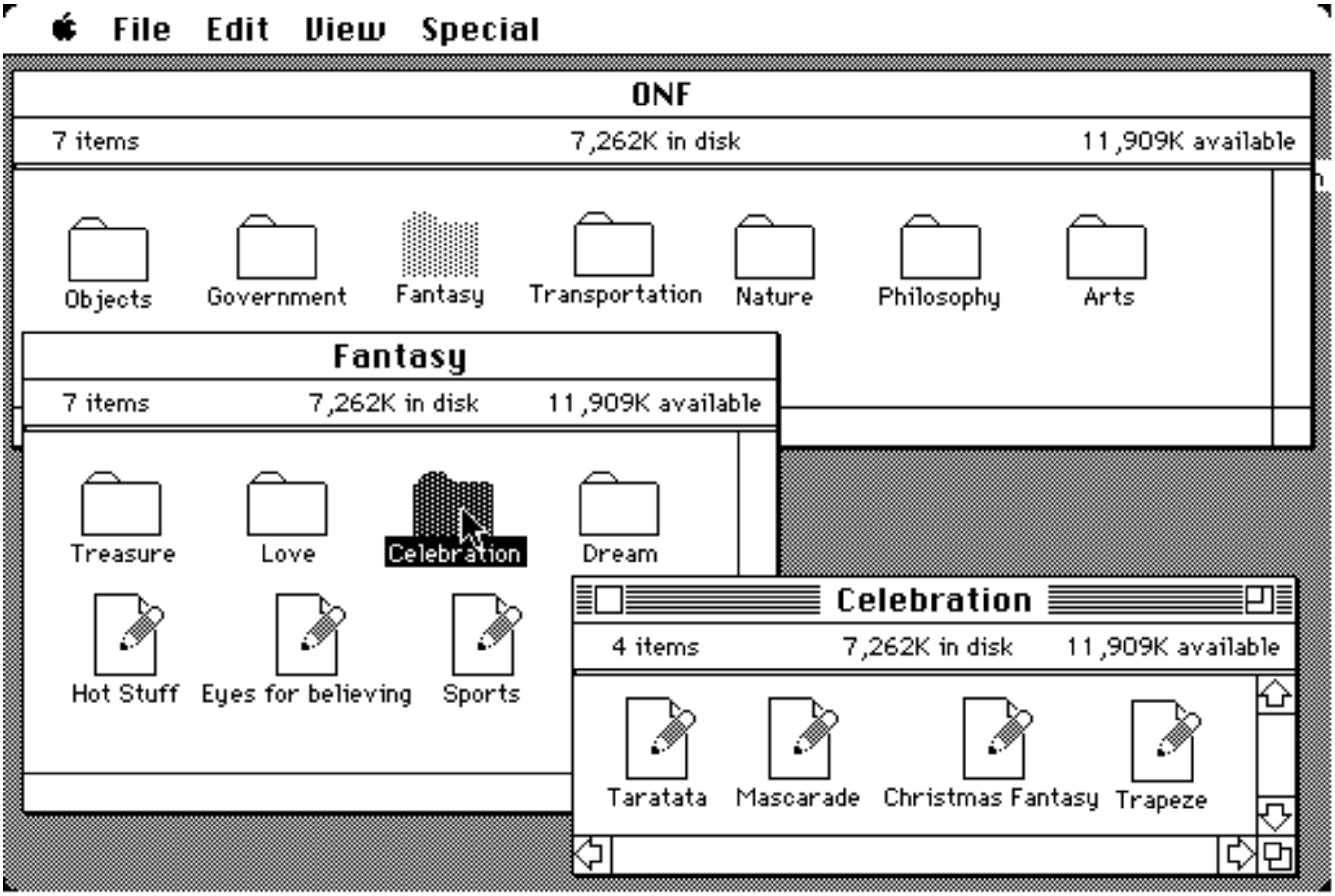


Figure 11. Sample screen from the MACINTOSH™ FINDER.

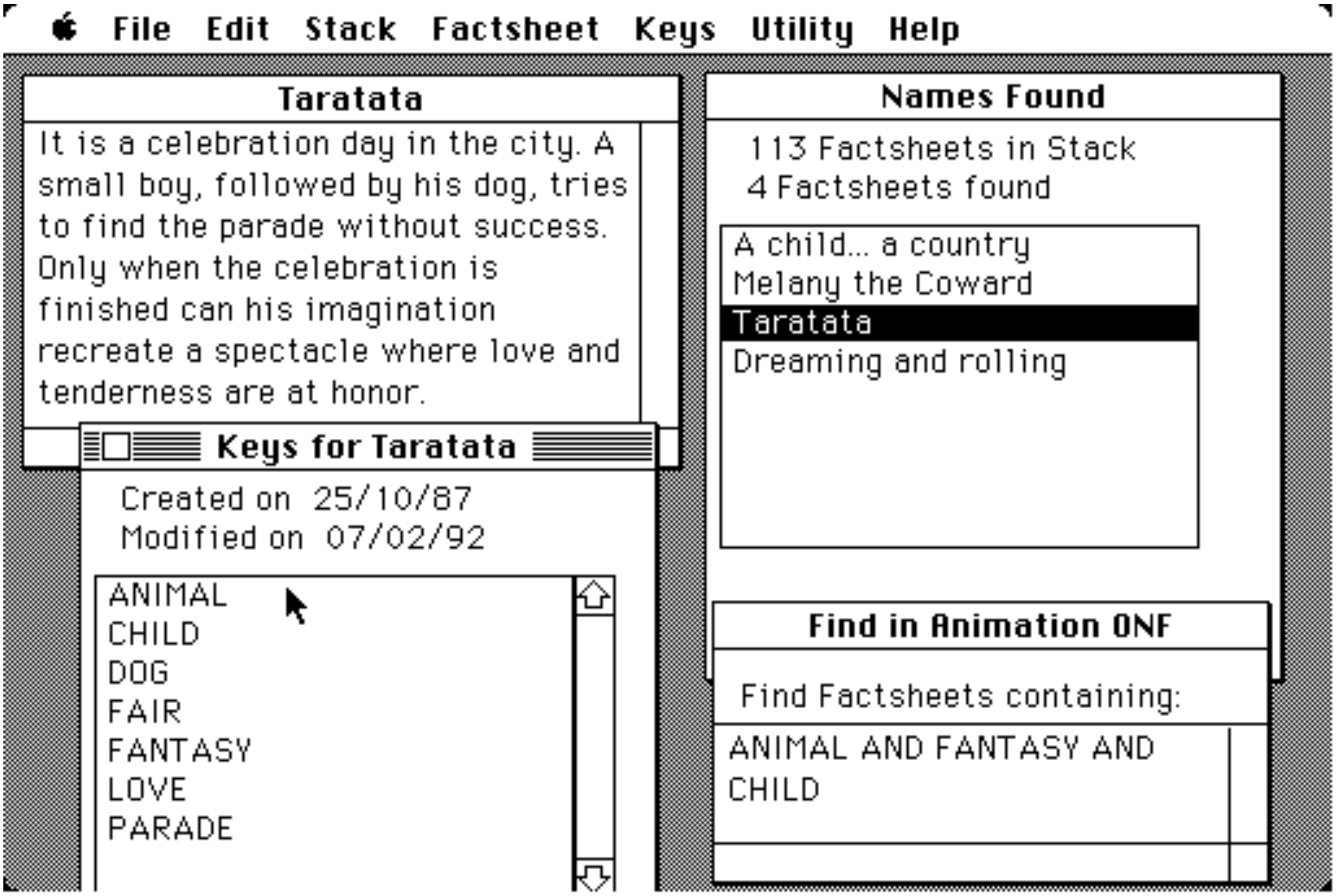


Figure 12. Example screen from the FACTFINDER™ system.