A system for conceptual structuring and hybrid navigation of text databases*

Claudio Carpineto and Giovanni Romano
Fondazione Ugo Bordoni
Via Baldassarre Castiglione 59, 00142, Rome, Italy
{carpinet, romano}@fub.it

Abstract

We present a three-stage system for information exploration and retrieval. Given a collection of texts, the first stage extracts a set of index terms describing each text. The second stage, named GALOIS, exploits the indices determined in the earlier stage to build a particular lattice of clustered texts. The resulting structure supports the third stage, a visual retrieval interface named ULYSSES. ULYSSES provides multiple and integrated retrieval strategies; the user may browse or query the clustered search space, or, in addition, he may restrict it by specifying constraints that the desired texts have to satisfy. The system, as a whole, has two main advantages. The first is that it is fully automatic and works in unrestricted environments. The second is that it combines browsing capabilities with good retrieval performance. In fact, the experiments conducted so far suggest that it can be seen as an alternative to more conventional systems even for retrieval tasks such as subject searching and for databases of reasonable size.

The organizing/navigating paradigm

One increasingly popular approach to information retrieval consists of organizing the database information into some structure and letting the user navigate through it (Thompson & Croft, 1989; Maarek and Berry, 1991; Bowman et al, 1994). The organizing/navigating approach has the chief advantage that it is also suitable for users who do not have a specific subject in mind or who do not know how to get it. In particular, it allows the user to retrieve documents of interest without specifying a query perfectly or partially matching some description of the documents. However, this approach presents also some limitations. One problem is the cost of building the support structure; creating the atomic pieces of information and linking them usually requires subjective and time-consuming decisions.

* Work carried out in the framework of the agreement between the Italian PT Administration and the Fondazione Ugo Bordoni

Another main concern is the retrieval effectiveness; the user may become disoriented or 'lost in hyperspace', and he may have to explore many links before finding the desired information. In fact, one central problem in navigation systems is to combine browsing potentials with good retrieval performance.

In this paper we address both these issues. We describe a system that provides a comprehensive approach to automatic structuring and hybrid navigation of text databases. The system architecture is displayed in figure 1. The first two components are responsible for building the structure supporting the text retrieval. In particular, the indexing component determines a set of index terms describing each text, while the clustering component applies to the resulting text by term matrix to build a network of clustered texts characterizing the whole collection. The third components is the actual retrieval interface to the network. In the next sections we describe the main components of the systems and their interactions. Also, we report the main result of an experimental evaluation of the system's retrieval effectiveness.

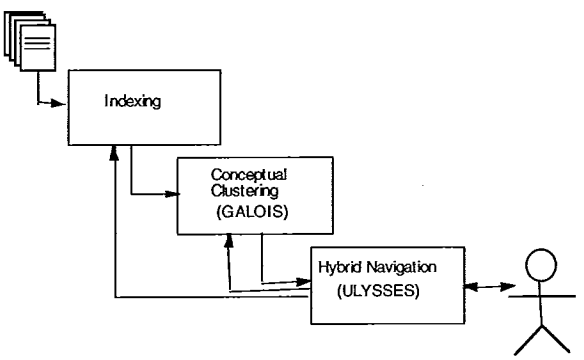

Figure 1. System's architecture

Automatic indexing of texts

The first task for building a conceptual representation of a text database is to identify the content of each text. Basically, there are two kinds of approaches. In the AI-based approach, natural language processing or machine learning techniques (Sowa, 1984; Srihari and Burhans, 1994; Baudin et al, 1994) can be used to build or refine an internal representation of each text.

Although most of these methods can produce *deep* conceptual indices, they can only work in restricted environments and usually require extensive knowledge about the semantics of the application domain. An alternative approach for content extraction that is domain-independent, mostly effective[1], and efficient is adopted in most information retrieval systems. Our indexing procedure is inspired by the latter approach. It consists of five steps: *text segmentation, word stemming, stop wording, word weighting, word selection.* At the end of this process each document is indexed by a restricted but informative set of indexes which is suitable for the subsequent processes of cluster formation and navigation. We applied this method to the data set CISI, a widely used bibliographical collection of 1460 information science abstracts. A detailed description of the indexing method as well as of its application to CISI is provided in (Carpineto and Romano, 1995b). As an illustration, we show in table 1 a small sample consisting of 6 documents described by 10 automatically determined index terms.

Table 1. A document by term matrix

	1	2	3	4	5	6
CLASSIFICATION				x	x	
CROSS-REFERENCE	x					
DOCUMENT		x	x	x		
INDEX	x		x		x	x
INFORMATION	x		x	x	x	
LANGUAGE	x					x
LIBRARY			x			
MEDLAR			x			
SYSTEM		x	x			x
THESAURUS	x	x		x		x

[1]It has been often reported that in many practical situations the use of more elaborated indexing systems than single-word term extraction did not result in significant improvement of retrieval performance (Salton, 1989).

Lattice conceptual clustering of the indexed database

The second stage of our system exploits the index terms determined in the earlier stage to build a concept network characterizing the whole database. The approach is based on a particular clustering structure, called concept (or Galois) lattice. Given a binary relation between a set of documents and a set of terms, the concept lattice is a set of clusters, in which each cluster is a couple, composed of a subset of documents (D), called extent, and a subset of terms (T), called intent. Each couple (D,T) must be a complete couple, meaning that T must contain just those terms shared by all the documents in D, and, similarly, the documents in D must be precisely those sharing all the terms in T. The set of couples can then be ordered by applying the standard set inclusion relation to the set of terms (or, dually, to the set of documents) that describe each couple. The resulting ordered set, which is usually represented by a Hasse diagram, turns out to be a lattice.

In figure 2 we show the Galois lattice of the simple database shown in table 1. The ascending paths represent the subclass/superclass relation; the bottom class is defined by the set of all terms and contains no documents, the top class contain all documents and is defined by their common terms (none, in this case). Note that, due to the completeness requirement, the lattice usually contains only a small subset of the set of classes that can be (theoretically) generated combining the terms in all possible ways.

Given the definition of concept lattices, we addressed the problem of their automatic determination. We implemented in a system named GALOIS an algorithm that builds the lattice incrementally, where each update takes time proportional to the number of documents to be clustered. We also studied the space complexity of Galois lattice, and found empirical and theoretical evidence that the size of the lattice grows linearly with respect to the number of documents. A detailed explanation of concept lattices, of their complexity and of the construction algorithm is contained in (Carpineto and Romano, 1993; Carpineto and Romano, 1994a).

In the next section we characterize the utility of concept lattices for supporting text retrieval, which has first been explored in (Godin et al, 1989).

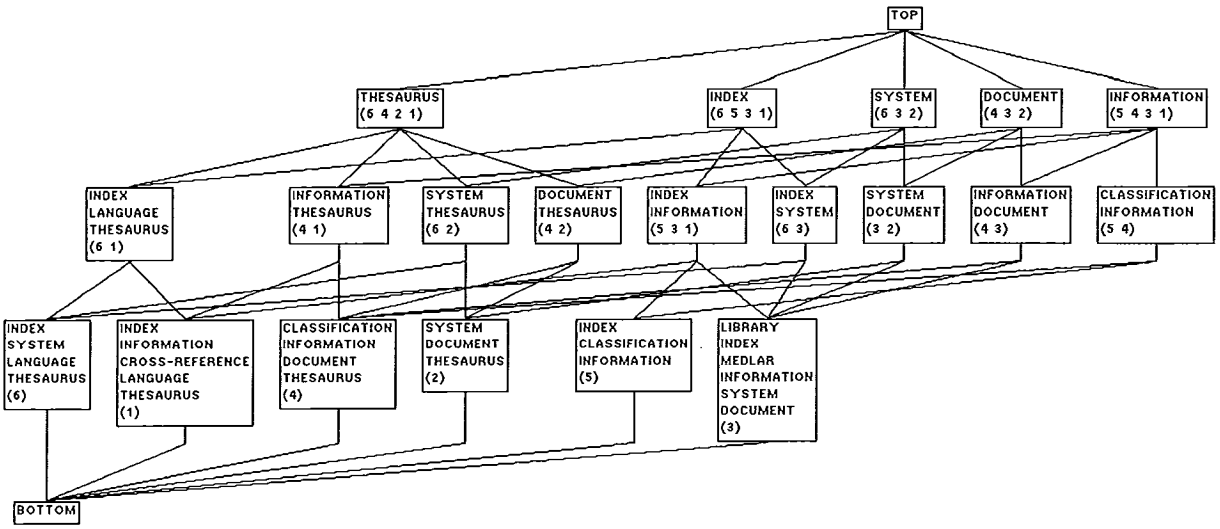

Figure 2. The concept lattice of the database in table 1.

Concept lattices as retrieval support structures

The potentials of clustering for information retrieval have long been known, the main justification for this being what van Rijsbergen termed the *cluster hypothesis*, namely the fact that documents associated in the same clusters tend to be relevant to the same questions. However, the statistical clustering methods that have been predominately been used in information retrieval (e.g., Willet, 1988; Crouch et al, 1989; Maarek et al, 1991) are usually affected by their inability of producing a conceptual description of the classes generated. By contrast, the clusters contained in a concept lattice have an intensional description, which may improve both the effectiveness of hierarchy navigation, for interactive searches, and the efficiency of the query-cluster matching process, for automatic searches. This seems to be one key feature for supporting browsing retrieval , but the task in question suggests other useful properties of the cluster structure.

- Graph navigation is more flexible than tree navigation. While in a strict hierarchical clustering each class has exactly one parent, in a lattice clustering there are many paths to a particular class. This facilitates recovery from bad decision making while traversing the hierarchy in search of documents.
- It is usually the case that a same document is relevant to two or more queries which happen to have incomparable descriptions. Therefore the ability to deal with non-disjoint classes is an important feature of browsing retrieval systems; lattice conceptual clustering naturally supports this functionality, as opposed to hierarchical conceptual clustering.
- In information retrieval domains there is usually an available body of background knowledge expressed as a thesaurus of terms. The ability to incorporate such knowledge into the clustering of documents may considerably improve retrieval performance.[2]
- Because text databases may be very large, the computational complexity involved in cluster hierarchy formation is of great importance. $O(n^2)$ time and $O(n)$ space clustering algorithms are generally considered to be efficient for retrieval purposes (Willet, 1988).
- In dynamic databases the structure underlying navigation may change as new items are added. Incremental conceptual clustering techniques may therefore be important to reduce the response time of the browser.

---

[2]The basic Galois lattice is a purely syntactic structure, in which the order over the classes is independent of possible semantic relationships between terms; however, in presence of auxiliary information expressed as a subconcept/superconcept relationship between the terms, the lattice can be adapted so that more general attributes index more general classes. The essence of this generalization is that when we compute the terms shared by sets of documents we have to take into account also the terms that are implicitly possessed by each document according to the auxiliary information. Indeed, GALOIS can build a thesaurus-enriched lattice (see Carpineto and Romano, 1994a).

A concept lattice meets all these requirements. Furthermore a concept lattice enjoys other useful properties, which make it suitable for supporting a hybrid navigational paradigm involving multiple and integrated retrieval strategies. The first is that in addition to supporting browsing, a concept lattice of documents also allows an easy form of direct query specification. In fact, each node in the lattice can be seen as a query formed of a conjunction of terms (the intent) with the retrieved documents (the extent). The second is that the lattice allows gradual enlargement or refinement of a query. More precisely, following edges departing upward (downward) from a query produces all minimal conjunctive enlargements (refinements) of the query with respect to that particular database. Third, the lattice supports a useful and simple form of incremental pruning of the search space driven by user-specified term-based constraints. In the next section we describe a visual retrieval interface based on these principles.

Hybrid navigation of lattice text representations

To enable the interaction between the user and the lattice we have realized a prototype interface on top of GALOIS, named ULYSSES[3]. The first problem in the design of the interface is the visualization of the search space. We adopted an approach similar to generalized fish-eye view (Furnas, 1986), in which there is a current focus of interest and the adjacent nodes are displayed with decreasing level of detail as we move away from the focus. As the lattice is typically too large to fit on a screen, we defined some parameters controlling the size and the topology of the region to be displayed. The advantage of this approach is that of selectively maximizing the amount of information that can be displayed, without sacrificing local detail around the focus. In figure 3 we show an example screen of ULYSSES relative to the lattice in figure 2, where the current focus is the highlighted node. The lattice is re-displayed whenever some action taken by the user has the effect of modifying the current focus or some of its neighboroughing nodes in the current screen.

The first way for the user to retrieve documents of interest is *browsing* through the network. ULYSSES allows selection of any node on the current screen by graphical direct manipulation, i.e. by pointing and clicking with the mouse on the desired node, and display of the documents associated with it. The second search strategy is *querying*. A query can be formulated in two manners: either the user specifies the new terms from scratch, or the user modifies the current query (i.e., the intent of the current focus). In the latter case the user can remove some terms, or add new terms, or generalise/specialize some current terms using the information contained in the thesaurus. The result of a query is the class of the lattice that exactly matches the query, if there is any, or one or more classes which best partially match the query according to some heuristic criteria. As a third interaction mode, ULYSSES allows the user to restrict the retrieval space from which he is retrieving information, rather than just accessing it in various ways (see feedback on the GALOIS module in figure 1). We called this *bounding*, because the user may specify constraints that the sought documents have to satisfy, and the search space is bounded accordingly using a representation similar to version spaces. The constraint are expressed as inequality relations between the description of any admissible class c and some conjunction of terms c_1. In our framework, which is logically similar to that proposed in (Mellish, 1991), there are four types of constraints (i.e., $c \geq c_1$, $c \leq c_1$, $c \neg\geq c_1$, $c \neg\leq c_1$); each constraint produces a simple graphical partition of the search space, and possesses a meaningful interpretation from an information retrieval point of view. As more and more constraints are seen the search space shrinks and may eventually converge to the target class; if the constraints are too strong the space becomes empty and the user is given the possibility of retracting some of the previously asserted constraints. The bound facility of ULYSSES is best described in (Carpineto and Romano, 1994b).

One of the most interesting features of ULYSSES is that these interaction modes can be naturally combined to produce a hybrid retrieval strategy that best reflects the user's goal and domain knowledge. For instance, the user may first query the system to locate the region of interest, and then browse through it; bounding may occur at any time, based on the knowledge that the user has or learns by the feedback from the structure. Compared to other hybrid approaches, in which the single strategies (e.g., browsing, query) typically work in parallel or in cascade, this represents a much tighter integration. In ULYSSES the different retrieval methods work on a same retrieval space and share their intermediate results; in this way, the user does not have to commit himself to one particular method and does not have to worry about mapping different representations and results, while the system does not have to maintain different structures. A thorough discussion of the whole interface and of the increased retrieval capabilities of hybrid approaches is contained in (Carpineto and Romano, 1995a)

[3]ULYSSES, like the indexing module and GALOIS, has been implemented in Common Lisp, and runs on a Symbolics Lisp Machine. We are currently porting the whole system, which consists of about 300 K-bytes of code, to PowerMac.

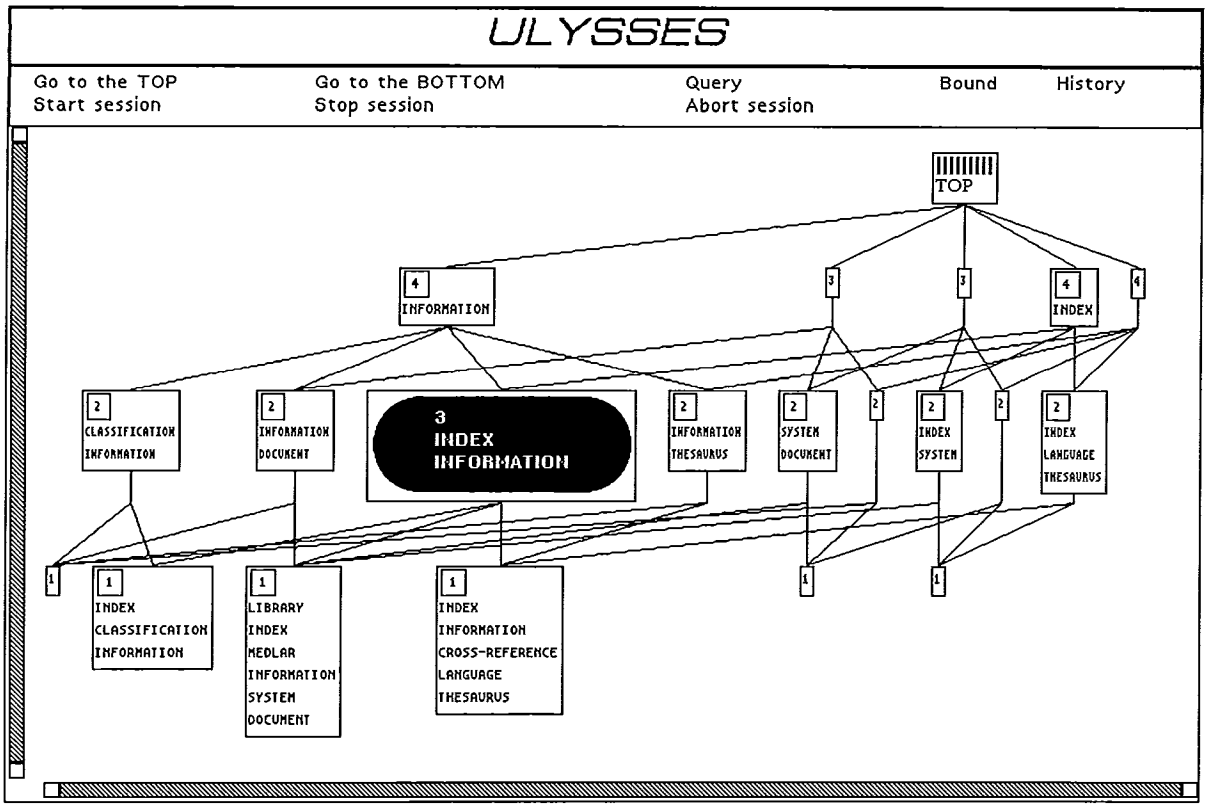

Figure 3. Display screen of ULYSSES, relative to the lattice in figure 2, focusing on the node INDEX, INFORMATION

Evaluation of the system's retrieval effectiveness

We have made several experiments to evaluate the retrieval effectiveness of the system, or of some of its components. Some of these results are reported in (Carpineto and Romano, 1994a). In a more recent experiment we compared the performance of the full system with that of a Boolean retrieval system on subject searching. The experiment was conducted on the CISI data set - a standard test collection containing 1,460 documents - that was first automatically indexed as specified in section 2. The same indexing relation was next used to generate the Boolean and the lattice databases. Afterwards, two external subjects were asked to retrieve the documents relevant to 10 of the 35 standard queries associated with the CISI data set. The lattice method showed significantly better recall and precision compared to the Boolean method. These results, which are best described in (Carpineto and Romano, 1995b) are particularly encouraging, because they seem to suggest that a lattice-based approach to information retrieval can be seen as an alternative to more conventional methods even for a typical retrieval tasks such as subject searching, and for reasonably-sized databases.

Conclusion

The system presented here brings artificial intelligence, information retrieval and user interface techniques to bear in text navigation and retrieval. The system has two main advantages. The first is robustness. It can be applied to any collection of texts, without subject matter, extent or scope restrictions and without the need for preconstructed knowledge structures. The second advantage of the system is flexibility. By providing the user with multiple and integrated search strategies, it combines browsing capabilities with good retrieval performance.

In order to expand the applicability of the system, we plan to make it available on the Internet as a tool for conceptual query refinement. The idea is that of using the powerful World Wide Web retrieval tools as a pre-processing step, and then applying our system to the retrieved information for improving its presentation and facilitating its access. Another, more theoretical, research direction is to investigate the possibility of working with scalable support structures. The idea is to work with a concept lattice in which the set of index terms may be restricted or enlarged automatically, based on the user goal and other possible design constraints (see

feedback on the indexing module in figure 1). We believe that this could provide a more a flexible way to trade off multiple conflicting objectives, such as maximizing efficiency and effectiveness and minimizing the complexity of interaction between the user and the system.

References

Baudin, C., Pell, B., Kedar, S. (1994). Incremental Acquisition of Conceptual Indices for Multimedia Design Documentation. *Proceedings of the AAAI-94 Workshop on Indexing and Reuse in Multimedia Systems,* Seattle, Washington.

Bowman, M., Danzig, P., Manber, U., & Schwartz, F. (1994). Scalable Internet Resource Discovery: Research Problems and Approaches. Communications of the ACM, 37, 8, pp. 98-114).

Carpineto, C., & Romano, G. (1993). GALOIS: An order-theoretic approach to conceptual clustering. Proceedings of the 10th International Conference on Machine Learning (pp. 33-40), Amherst, MA:Morgan Kaufmann.

Carpineto, C., & Romano, G. (1994a). A lattice conceptual clustering system and its application to browsing retrieval. Accepted for publication in Machine Learning.

Carpineto, C., & Romano, G. (1994b). Dynamically bounding browsable retrieval spaces: an application to Galois lattices. In Proceedings of RIAO 94: Intelligent Multimedia Information Retrieval Systems and Management(pp. 520-533), New York.

Carpineto, C., & Romano, G. (1995a). ULYSSES: A lattice-based multiple interaction strategy retrieval interface. In Proceedings of EWHCI'95: 5th East-West Human Computer Interaction Conference, Moscow .

Carpineto, C., & Romano, G. (1995b). Automatic construction of navigable concept networks characterizing text databases. To appear in Proceedings of the Fourth Congress of the Italian Association for Artificial Intelligence, Florence, Italy.

Crouch, D., Crouch, C., & Andreas, G. (1989). The use of cluster hierarchies in hypertext information retrieval. Proceedings of the ACM Hypertext '89 Conference (pp. 225-237), Pittsburgh, PA: ACM.

Furnas, G. (1986). Generalized fisheye views. Proceedings of the Human Factors in Computing Systems (pp.16-23). North Holland.

Godin, R., Gecsei, J., & Pichet, C. (1989). Design of a browsing interface for information retrieval. Proceedings of the Twelfth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 32-39). Cambridge, MA: ACM Maarek, Y., Berry, D., & Kaiser, G. (1991). An Information Retrieval Approach For Automatically Constructing Software Libraries. IEEE Transactions on Software Engineering, 17, 8, 800-813.

Mellish, C. (1991). The description identification problem. Artificial Intelligence, 52, 2, 151-168.

Salton, G. (1989). Automatic Text Processing: The transformation, Analysis and Retrieval of Information by Computer. Addison Wesley.

Sowa, J. (1984). Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley, 1984.

Srihari, R., Burhans, D. (1994). Visual Semantics: Extracting Visual Information from Text Accompanying Pictures. Proceedings of AAAI-94, Seattle, Washington, AAAI Press.

Thompson, R., & Croft, B. (1989). Support for browsing in an intelligent text retrieval system. International Journal of Man-machine Studies, 30, 639-668.

Willet, P. (1988). Recent trends in hierarchic document clustering: a critical review. Information Processing & Management, 24, 5, 577-597.