Automatic Construction of Navigable Concept Networks Characterizing Text Databases

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

Abstract

In this paper we present a comprehensive approach to conceptual structuring and intelligent navigation of text databases. Given any collection of texts, we first automatically extract a set of index terms describing each text. Next, we use a particular lattice conceptual clustering method to build a network of clustered texts whose nodes are described using the index terms. We argue that the resulting network supports an hybrid navigational approach to text retrieval - implemented into an actual user interface - that combines browsing potentials with good retrieval performance. We present the results of an experiment on subject searching where this approach outperformed a conventional Boolean retrieval system.

1. Introduction

Especially after the advent of Internet, the amount of available information represented as text databases has rapidly grown. This is the case for bibliographic collections, news and message files, software libraries, multimedia repositories with text captions, etc. Most usually, these databases are accessed to retrieve items of interest or to explore their content, therefore it is useful to organize the information into some sort of structure and then let the user navigate through it. In fact, the organizing/navigating paradigm is becoming very popular (Thompson and Croft, 1989; Maarek et al, 1991; Lucarella et al, 1993; Bowman et al, 1994). One main problem of this approach is creating the atomic pieces of information and linking them, which usually requires subjective and time-consuming decisions. We argue that conceptual clustering techniques (Michalski, 1983) may be helpful to automate this process.

In an abstract browsing retrieval setting there is a collection of documents described by a set of index terms and the user wants to find elements of interest by navigating through a network of clustered documents. Conceptual clustering presents three basic features for supporting this task, in that the clusters are (a) automatically generated from the document-term relation, (b) described by a conceptual description, (c) organized into a hierarchy showing their generality/specificity. In this paper we present an approach to organizing a collection of documents that is based on a particular lattice conceptual clustering method (Carpineto and Romano, 1993). It turns out that the resulting clustering structure presents many useful advantages for supporting browsing retrieval from the document collection, but in order to build a practical system two more questions need be addressed. The first concerns the set of index terms used in the clustering process. While in most conceptual clustering task it is assumed that the set of properties describing the objects to be clustered is given, this assumption is often unrealistic for textual objects, which are usually available unindexed. To solve this problem we have developed a module for automatic indexing that combines linguistic and statistical concepts. The second issue to address is the construction of an actual interface to enable the interaction between the user and the support network. We have realized a visual interface to the concept lattice that provides a hybrid navigational strategy integrating two conventional retrieval techniques - link-based and query-based navigation - and a novel one - user-bounded navigation.

A major part of this paper is an empirical evaluation of the retrieval effectiveness of the overall system. It is often remarked that, at least for sizable databases and typical retrieval tasks such as subject searching, the structure-based approaches to information retrieval have not yet proven to be an effective alternative to more conventional query-based methods. We address this issue, and present the results of an experiment on subject searching in a standard reasonably-sized database, where the lattice method performed significantly better than a conventional Boolean system.

The rest of the paper is organized in the following way. In section 2 we present the automatic indexing method. In section 3 we introduce the lattice conceptual clustering method, while in section 4 we characterize its utility for browsing retrieval. Section 5 contains a description of the visual retrieval interface. In section 6 we describe the experimental evaluation. Finally, we offer some conclusion and directions for future work in section 7.

2. Automatic Indexing

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 techniques (Sowa, 1984; Srihari and Burhans, 1994) or machine learning techniques (Barletta and Mark, 1988; 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, knowledge-free, and efficient is adopted in most information retrieval systems. Our indexing procedure is inspired by the latter approach. It consists of the following steps.

1. Text segmentation

Our system first identifies the individual words occurring in a text collection, ignoring punctuation and case.

2. Word stemming

We reduce each word to word-stem form. This is done by using a very large morphological lexicon for English (Karp et al, 1992) that contains the standard inflections for nouns (singular, plural, singular genitive, plural genitive), verbs (infinitive, third person singular, past tense, past participle, progressive form), adjectives (base, comparative, superlative). The lexicon can handle more than 317,000 inflected forms derived from over 90,000 roots.

3. Stop wording

We use a stop list to delete from the texts the (root) words that are insufficiently specific to represent content. Our stop list has been obtained by unioning two stop lists, a "pragmatic" one and a "grammatical" one. The former is a typical stop list in use in information retrieval, included in the CACM dataset; it contains 428 common function words, such as *the*, *of*, *this*, *on*, etc. and some verbs, e.g., *have*, *can*, *indicate*, etc. The latter, consisting of 403 words, was built by extracting from the lexicon mentioned above all the entries labeled as either pronouns, or conjunctions, or prepositions, or determiners. The final stop list contains 638 words.

4. Word weighting

For each remaining word we derive a measure of its usefulness for indexing purposes. The goal is to identify words that characterize the documents to which they are assigned, while also discriminating them from the remainder of the collection. In fact, most of the weigthing functions that have been proposed tend to favor terms occurring with relatively high frequencies in some individual documents, but with a relatively low overall collection frequency (Salton, 1989). We use the *Signal-Noise Ratio* method, which has an information theory basis; the signal of term k for a collection of n documents is defined as follows

$$SIGNAL_k = \log_2(TOTFREQ_k) - NOISE_k \quad \text{, where}$$

$$NOISE_k = \sum_{i=1}^{n} \frac{FREQ_{ik}}{TOTFREQ_k} \log_2 \frac{TOTFREQ_k}{FREQ_{ik}}$$

5. Index word selection

Each document is indexed by the set of words relative to the document whose signal-noise ratio is greater than a fixed threshold; if some document remains unindexed, it is assigned a minimum fixed number of words with the highest scores. By varying these two parameters we can choose an informative but restricted set of indexes which is suitable for the subsequent processes of cluster formation and visualization.

This basic indexing method could be improved in a variety of ways, without hurting its generality and efficiency. For instance, we could use vocabulary normalization tools such as synonym list and thesarus associations, or using adjacent words to form term-phrases (Maarek et al, 1991; Chen et al, 1994). We have to emphasize, however, that it has been often reported that in many practical situations the use of more elaborated systems than single-word term extraction did not result in significant performance improvement (Salton, 1989).

Computationally, the indexing module works fairly well. We organized the lexicon into a digital search tree, or trie, and represented all other information (stop-words, weights, etc.) into the same stucture. The memory requirement is still quite demanding (about 10 M-bytes, approximately as much as that of the original unstructured lexicon) but the whole indexing process is fast.

We tested the system on the data set CISI, a widely used bibliographical collection of 1460 information science abstracts. The data set initially contained 184,584 words, 9706 of which were distinct. After stemming and stop wording the data set contained 90154 words, with 7598 distinct words. We next computed the signal-noise ratio and selected all words whose weight was greater than or equal to 1.00, the minimum number of indices per document being set to 4. With this choice each document was indexed by an average of 5.2 terms. To illustrate this application, in table 1 we show a small CISI sample consisting of 6 documents. The table also shows the set of indices associated with each document, with their signal-noise ratio values.

Table 1. An automatically indexed subset of the CISI collection

---

Doc. 1
Vocabulary Building and Control Techniques, Wall, Eugene
The rationale is given for creation and maintainance by an information center of a controlled indexing and retrieval vocabulary.. Basic vocabulary principles are (1) use of natural language, (2) development of hospitality to new concepts, (3) provision of adequate cross-referencing, and (4) formatting for easy use.. Terminalogical conventions necessary for development and control of a useful vocabulary are summarized, and the techniques for applying these conventions to construct a thesaurus are described.. Computerized editing techniques and updating techniques are briefly set forth..
(INDEX 1.85) (INFORMATION 1.64) (CROSS-REFERENCE 1.63) (LANGUAGE 1.44) (THESAURUS 1.43) (VOCABULARY 1.26) (RETRIEVAL 1.26) (CENTER 1.03)

Doc. 2
Word-Word Associations in Document Retrieval Systems, Lesk, M. E.
The SMART automatic document retrieval system is used to study association procedures for automatic content analysis.. The effect of word frequency and other parameters on the association process is investigated through examination of related pairs and through retrieval experiments.. Associated pairs of words usually reflect localized word meanings, and true synonyms cannot readily be found from first or second order relationships in our document collections.. There is little overlap between word relationships found through associations and those used in thesaurus construction, and the effects of word associations and a thesaurus in retrieval are independent.. The use of associations in retrieval experiments improves not only recall, by permitting new matches between requests and documents, but also precision, by reinforcing existing matches.. In our experiments, the precision effect is responsible for most of the improvement possible with associations.. A properly constructed thesaurus, however, offers better performance than statistical association methods..
(SYSTEM 1.63) (DOCUMENT 1.60) (THESAURUS 1.43) (WORD 1.38) (REQUEST 1.34) (RETRIEVAL 1.26) (SMART 1.25) (PRECISION 1.18) (AUTOMATIC 1.16)

Doc. 3
A Comparison Between Manual and Automatic Indexing Methods, Salton, Gerard
The effectiveness of conventional document indexing is compared with that achievable by fully automatic text processing methods.. Evaluation results are given for a comparison between the MEDLARS search system used at the National Library of Medicine and the experimental SMART system, and conclusions are reached concerning the design of future automatic information systems..
(LIBRARY 2.12) (INDEX 1.85) (MEDLAR 1.81) (INFORMATION 1.64) (SYSTEM 1.63) (DOCUMENT 1.60) (SEARCH 1.38) (SMART 1.25) (TEXT 1.24) (AUTOMATIC 1.16)

Doc. 4
Automated Keyword Classification for Information Retrieval, Sparck-Jones, K.
This book is primarily a research monograph, in which the discussion of the main topics has been broadened so that they are related to their surrounding context in information retrieval as a whole; it is not a textbook, and no attempt has therefore been made to justify the choice of topic, or account for the use of certain concepts, or to provide an elementary description of either. For instance in Chapter 1, it is assumed that the reader is familiar with the idea of using keywords in information retrieval: I have not considered the relation between this kind of retrieval device and a controlled thesaurus or descriptor set, or that between the use of simple class lists as document descriptions and the use of descriptions with a syntactic structure, for example. Equally, in Chapter 2, I have made use of recall/precision ratios as a means of characterising retrieval performance, without justification or argument; but this does not mean that I am unaware of the difficulties of doing this, or of the attention which has been devoted to, and controversy which has raged round, this subject; it is simply that from the point of view of my main purpose it is reasonable to use these ratios.
(CLASSIFICATION 1.74) (INFORMATION 1.64) (DOCUMENT 1.60) (THESAURUS 1.43) (RETRIEVAL 1.26) (DESCRIPTOR 1.26) (CHAPTER 1.24) (BOOK 1.20) (PRECISION 1.18) (SUBJECT 1.02)

Doc. 5
Classification and Indexing in Science, Vickery, B.C.
The preface to the first edition of this book - which is reproduced following this - shows that in 1958 the classification ideas in it were felt to be controversial, needing to be championed. A few years before, the Classification Research Group had issued a memorandum proclaiming 'the need for a faceted classification as the basis of all methods of information retrieval.' As part-author of this memorandum, I must now judge the claim to have been too bold, even brash. But it has been vindicated to an extent, for both in theory and practice the value of facet analysis, in the organization of subject vocabularies for indexing and search, has been widely accepted - whether these vocabularies are classified or alphabetical, and whether used in pre- or post-coordinate fashion.
(INDEX 1.85) (CLASSIFICATION 1.74) (INFORMATION 1.64) (SEARCH 1.38) (VOCABULARY 1.26) (SCIENCE 1.26) (RETRIEVAL 1.26) (BOOK 1.20) (THEORY 1.15) (SUBJECT 1.02)

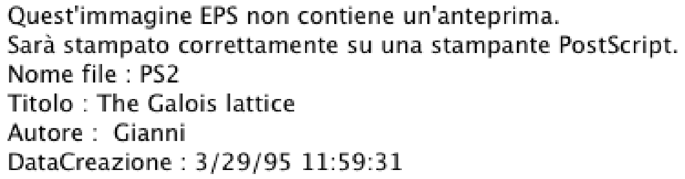
Doc. 6
Factors Determining the Performance of Indexing Systems, Cleverdon, C.W.
The test results are presented for a number of different index languages using various devices which affect recall or precision. Within the environment of this test, it is shown that the best performance was obtained with the group of eight index languages which used single terms. The group of fifteen index languages which were based on concepts gave the worst performance, while a group of six index languages based on the Thesaurus of Engineering Terms of the Engineers Joint Council were intermediary. Of the single term index languages, the only method of improving performance was to group synonyms and word forms, and any broader groupings of terms depressed performance. The use of precision devices such as links gave no advantage as compared to the basic device of simple coordination.
(INDEX 1.85) (SYSTEM 1.63) (LANGUAGE 1.44) (THESAURUS 1.43) (TERM 1.40) (WORD 1.38) (PRECISION 1.18)

---

3. Lattice Conceptual Clustering

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 Galois lattice. Given a binary relation between a set of documents and a set of terms, the Galois 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.

We explain the structure by an example. Consider again the simple database shown in table 1 but, for the sake of illustration, assume that each document is described only by the terms whose score is > 1.4. The corresponding Galois lattice is shown in figure 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.



Quest'immagine EPS non contiene un'anteprima.
Sarà stampato correttamente su una stampante PostScript.
Nome file : PS2
Titolo : The Galois lattice
Autore : Gianni
DataCreazione : 3/29/95 11:59:31

Figure 1. The Galois lattice of the database in table 1.

Given the definition of Galois 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

Galois lattices, of their complexity and of the construction algorithm is contained in (Carpineto and Romano, 1994a).

Galois lattices have many applications, including prediction of unknown attributes and discovery of implications between database attribute values (Carpineto and Romano, 1993). In this paper we are interested in their utilization for supporting information retrieval. This issue is addressed in the next section.

4. 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 predominately been used in information retrieval (e.g., Willet, 1988; Crouch et al, 1989; Maarek et al, 1991) are affected by their inability of producing a conceptual description of the classes generated. By contrast, conceptual clustering techniques usually provide an intensional description of the clusters generated, 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 indeed one key feature for supporting browsing retrieval, but the task in question suggests other desirable 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.[1]

- 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

[1] 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).

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.

Most (conceptual) clustering systems can handle some of these issues but not all of them, GALOIS meets all these general requirements; in addition, a Galois 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.

5. Hybrid Navigation of Concept Lattices

To enable the interaction between the user and the lattice we have realized a prototype interface on top of GALOIS, named ULYSSES[2]. 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 2 we show an example screen of ULYSSES relative to the lattice in figure 1, 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 items of interst 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,

[2] 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.

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 she is retrieving information, rather than just accessing it in various ways. 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 c1. In our framework, which is logically similar to that proposed in (Mellish, 1991), there are four types of constraints (i.e., $c \geq c1$, $c \leq c1$, $c \neg\geq c1$, $c \neg\leq c1$); 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.

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. The bound facility is best described in (Carpineto and Romano, 1994b), while a thorough discussion of the whole interface and of the increased retrieval capabilities of hybrid approaches is contained in (Carpineto and Romano, 1995).

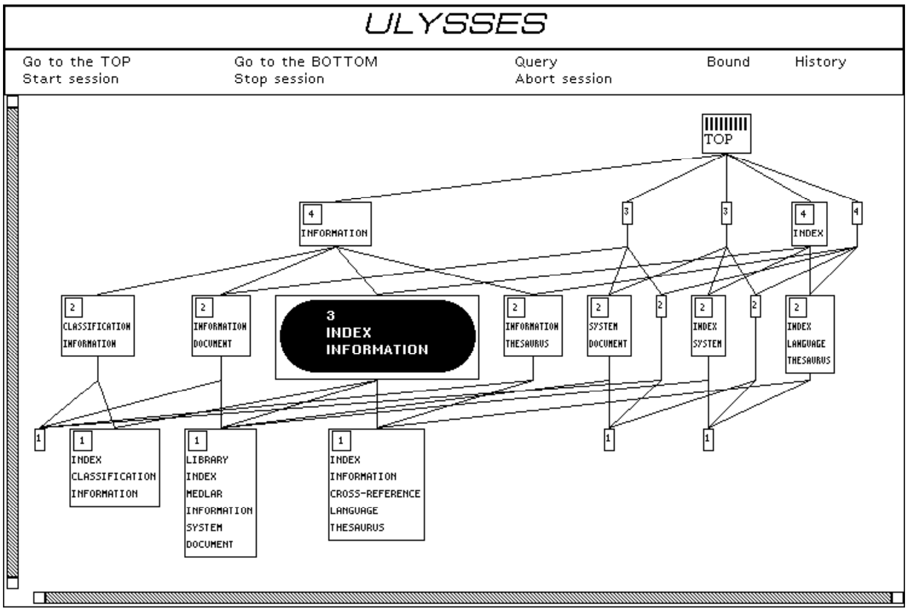


Figure 2. Display screen of ULYSSES, relative to the lattice in figure 1, focusing on the node INDEX, INFORMATION

6. Evaluation of the System's Retrieval Effectiveness

We evaluated the effectiveness of our system on subject searching; this is a familiar task where a user is to retrieve a set of documents relevant to a given question expressed in natural language. We compared the performance of our system with that of a Boolean retrieval system, which is known to perform well on this task and is easy to implement. The experiment was conducted on the CISI collection, 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. The lattice contained 4707 nodes described on average by 3.61 terms, with an average of 3.45 parents per node and a depth ranging from 2 to 11 edges.

The experimental protocol was as follows. The two retrieval methods were evaluated by two external subjects on 10 queries randomly selected among the 35 questions that are associated with the CISI data set[3]. Each question had its set of relevant documents associated with it, the average number of relevant documents per question being 39. For assigning questions to the two methods a repeated-measures design was used, in which each user searched each question using each method; also, to minimize sequence effects the experimentation of one method was did only a few days after the experimentation of the other, and the order of searching was varied over the question set. During each search the user, who was not asked to finish within a certain time period, could see the abstract of the documents returned in response to boolean queries - in the Boolean method - or associated to the nodes visited - in the lattice method. The documents judged to be relevant by the user, as well as those scanned during the search, were noted as retrieved documents. For each search we considered four measures: recall, precision,[4] number of queries formulated, search time (i.e., the time taken by the user to perform his task). The results are displayed in table 2.

Table 2. Average values of retrieval performance measures

Method	recall	precision	number of queries	search time (sec)
Boolean retrieval	0.32 ($\sigma = 0.16$)	0.42 ($\sigma = 0.15$)	5.87 ($\sigma = 1.35$)	1787 ($\sigma = 561$)
Lattice-based retrieval	0.45 ($\sigma = 0.18$)	0.58 ($\sigma = 0.10$)	2.00 ($\sigma = 0.75$)	1677 ($\sigma = 796$)

The table shows that the lattice-based method obtained better evaluation scores for

[3] An example of question is the following: What possibilities are there for automatic grammatical and contextual analysis of articles for inclusion in an information retrieval system?

[4] Recall is defined as the ratio of number of items retrieved and relevant to number of items relevant; precision is the ratio of number of items retrieved and relevant to number of items retrieved. Recall measures the ability to retrieve all relevant documents, while precision measures the ability to retrieve only relevant documents.

each measure. A paired t-test revealed no effect of the method on search time (p = 0.19), but it did reveal the superiority of the lattice method with respect to recall (p = 0.003) and precision (p = 0.07). Thus, these results show that the conceptual structuring of the database helped the user focus the search on relevant regions of the retrieval space, which resulted in an improvement of both recall and precision. More specifically, we observed that in the Boolean method it was often difficult to formulate queries returning document lists of reasonable size; consequently, the user was often engaged in time-consuming scannings of the vocabulary window in an attempt to refine overly general queries. By contrast, in the lattice-based method the user exploited the feedback obtained from the structure for refining direct queries, thus reducing the need for other direct queries. This phenomenon is also witnessed by the statistics on the number of queries, which are markedly different in the two methods. Before closing this section, we have to emphasize that this experiment was not designed to measure browsing capabilities, for which our approach has a clear advantage over the Boolean method.

7. Conclusion

The major contribution of this work consists of bringing artificial intelligence, information retrieval, and user interface techniques to bear to build intelligent and robust text retrieval systems. We first presented a method for automatically constructing concept networks characterizing unindexed text databases. We next discussed the potentials of this kind of structure for supporting text retrieval, and presented an actual interface for navigating through the network. The system described here is applicable to collections of texts that vary in subject matter, scope and extent, and does not require domain-specific preconstructed knowledge structures. We compared the performance of this approach on subject searching with that of a Boolean retrieval system. The results were quite encouraging, in that the lattice method showed significantly better recall and precision compared to the Boolean method.

The next step of this research will be aimed at expanding the applicability of the system by making it available on the Internet. The idea is that of using the powerful tools available in World Wide Web for searching networked information such as Netscape or Lycos, as a pre-processing step, and then applying our system to the set of elements retrieved in this way, specified in some fixed format. In this view, our system will act as a conceptual browser of the information retrieved after a Web search, for improving its presentation and facilitating its access.

Acknowledgments

This work has been carried out within the framework of the agreement between the Italian PT Administration and the Fondazione Ugo Bordoni.

References

Barletta, R., Mark, W. (1988). Explanation-Based Indexing of Cases. *Proceedings of AAAI-88*, St. Paul, Minnesota, Morgan Kaufmann.

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. Submitted to 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. (1995). ULYSSES: A lattice-based multiple interaction strategy retrieval interface. To appear *in Proceedings of EWHCI'95: 5th East-West Human Computer Interaction Conference*, Moscow .

Chen, H., Hsu, P., Orwig, R., Hoopes, L., Nunamaker, J. (1994). Automatic concept classification of text from electronic meeting. *Communications of the ACM*, 37, 10, pp. 57-73.

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.

Karp, D., Schabes, Y., Zaidel, M., Egedi, D. (1992). A freely available wide coverage morphological analyzer for English. *Proceedings of the 14th International Conference on Computational Linguistics (COLING '92),* Nantes, France.

Lucarella, D., Parisotto, S., Zanzi, A. (1993). MORE: Multimedia Object Retrieval Environment. *Proceedings of the Fifth ACM Conference on Hypertext* (pp. 39-50). Seattle, WA.

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.

Michalski, R., Stepp, R. (1983). Learning from observation: Conceptual clustering. In R. Michalski, J. Carbonell, T. Mitchell (Eds.*), Machine Learning: An Artificial Intelligence Approach (Vol. 1)*. Palo Alto, CA: Tioga Publishing.

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.