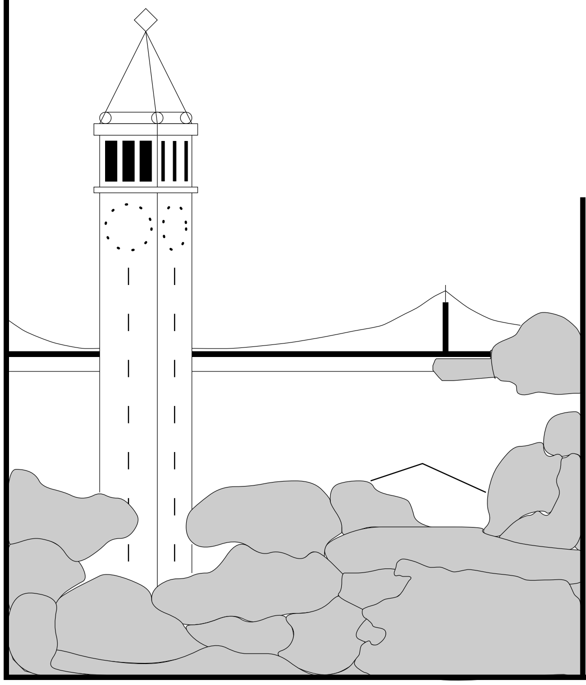
# Contextualizing Retrieval of Full-Length Documents

*Marti A. Hearst*



**Report No. UCB/CSD 94/789**
January 1994

Computer Science Division (EECS)
University of California
Berkeley, California 94720

UC Berkeley Computer Science Technical Report Number UCB/CSD-94-789

# Contextualizing Retrieval of Full-Length Documents

Marti A. Hearst
Computer Science Division, 571 Evans Hall
University of California, Berkeley
Berkeley, CA 94720
and
Xerox Palo Alto Research Center
*marti@cs.berkeley.edu**

**Abstract**

We address some issues relating to retrieval from unfamiliar text collections consisting of full-length documents. We claim that displaying query results in terms of inter-document similarity is inappropriate with long texts, and suggest instead that the results of simple initial queries should be contextualized according to category sets that correspond to the main topics of the texts. We argue that main topics of long texts should be represented by multiple categories, since in most cases one category cannot adequately classify a text. We describe a new automatic categorization algorithm that does not require pre-labeled texts and a prototype browsing interface that presents a simple mechanism for displaying multi-dimensional information.

## 1 Introduction

The recent proliferation of networked on-line text collections is heightening the need for information access systems that allow users to quickly and easily orient themselves to new datasets. Information retrieval research should support a paradigm in which it is easy for a user scanning over multiple datasets to issue a very simple query initially, get some idea of what kind of information is in the dataset being searched, and then either choose a different collection or reissue a more complex query that better fits the dataset. As (Croft & Das 1990) point out – relevance feedback, although a very useful tool, does not help with the initial search, and this initial search is time-critical for users of networked text collections.

Simple keyword queries can be composed quickly, but they tend to be either too general or too specific. When too general, the query is underspecified and the user must wade through a daunting number of documents. When too specific, no documents are returned. The problem of inappropriate search terms is exacerbated when users are unfamiliar with the text collection.

Researchers have suggested remedying this by improving the query. For example, I$^3$R (Croft & Thompson 1987) emphasizes detailed interaction with the user to improve the query. Systems based on the vector space model (Salton 1988) work best when provided with a verbose query, preferably a sample document from the space being searched. Neither of these solutions is satisfactory for the search situation we are concerned with here; they

*This research was sponsored in part by the Advanced Research Projects Agency under Grant No. MDA972-92-J-1029 with the Corporation for National Research Initiatives (CNRI).

require the user to transmit a relatively large amount of information to a source whose characteristics are unfamiliar.

The alternative we explore here is a paradigm in which the system expects simple queries and provides an effective way for their results to be browsed. More specifically, we suggest allowing users to issue simple keyword queries to systems that contextualize the results with category information. In effect, the system uses topic categories to characterize the documents that contain the search terms.

This paper takes on the following format. Section 2 discusses why inter-document similarity, the standard way to compare documents and display the results of retrieval, is inappropriate for full-length texts. The alternative we suggest is the use of category information, and Section 3 describes two ways in which multiple category information can be useful for contextualizing the results of retrieval with simple queries. Section 4 describes a new algorithm for automatically assigning multiple categories to full-length texts. These ideas are tied together in Section 5, which describes an implemented browsing interface that allows users to control what configuration of category information is displayed. Section 6 summarizes the themes of the paper.

## 2 Drawbacks of Comparing Full-Length Texts

Most information retrieval systems use inter-document similarity to compare documents and the results of retrieval. For example, the vector space model of similarity search (Salton 1988), categorization algorithms that use clustering to indicate document similarity (e.g., (Cutting *et al.* 1992), (Griffiths *et al.* 1986)), and latent semantic indexing for determining inter-document similarity (e.g., (Deerwester *et al.* 1990), (Chalmers & Chitson 1992)) all work by comparing the entire content of a document against the entire contents of other documents.

These modes of comparison are appropriate on abstracts because most of the (non-stopword) terms in a short text are salient for retrieval purposes, in part because they act as placeholders for multiple occurrences of those terms in the original text, and because generally these terms pertain to the most important topics in the text. When short documents are compared via the vector-space model or clustering, they are positioned in a multi-dimensional space where the closer two documents are to one another, the more topics they are presumed to have in common. This is often reasonable because when comparing abstracts, the goal is to discover which pairs of documents are most alike. For example, a query against a set of medical abstracts which contains terms for the name of a disease, its symptoms, and possible treatments is best matched against an abstract with as similar a constitution as possible.

A problem with applying standard information retrieval methods to full-length text documents is that the structure of full-length documents is quite different from that of abstracts. One way to view an expository text is as a sequence of subtopics set against a "backdrop" of one or two main topics. The main topics of a text are discussed in its abstract, if one exists, but subtopics usually are not mentioned. Being able to search full texts allows users to retrieve documents that contain only short discussions of a subject of interest. The problem that can accompany this is that passing references or short subtopical discussions are then returned even in cases where the user only wanted the document if the subject of interest is discussed at some length (i.e., as a main topic).

Most long texts discuss several main topics simultaneously; thus, two texts with one shared main topic will often differ in their other main topics. Some topic co-occurrences are more common than others; e.g., terrorism is often discussed in the context of U.S. foreign policy with the Middle East, and these two themes might even be grouped together in some domain-specific ontologies. However, texts often discuss themes that would not usually be considered to be in the same semantic frame; for example, (Morris 1988) includes an article that describes terrorist incidents at Bolshoi ballet performances. Thus classifying documents at a particular point in a topic hierarchically can be misleading.

Therefore, we hypothesize that algorithms that successfully group short texts according to their overall similarity (e.g., clustering algorithms, vector space similarity, and LSI), will produce less meaningful results when applied to full-length texts.

This hypothesis is supported by the fact that recently researchers experimenting with retrieval against datasets consisting of long texts have been breaking the texts into subparts, usually paragraphs, and comparing queries against these isolated pieces (e.g., (Salton *et al.* 1993), (Salton & Buckley 1992)). Presumably they found that matching a query against the entirety of a long text is less successful than matching against individual pieces. However, matching a query against paragraphs requires that a relevant document have at least one critical paragraph containing all of the salient query terms. (Hearst & Plaunt 1993) show that overall retrieval results against long documents can be improved by comparing the query against a combination of the best parts of each document. We infer from this that often the information relevant to the query appears in different pieces of the document. These results lend support to the claim that the structure of full-length texts should be taken into account when performing similarity comparisons.

In summary, we hypothesize that when long documents are clustered according to how similar they are throughout, it can be difficult to discern why they were grouped together if this grouping is a a function of some difficult-to-interpret intermediate position in multi-dimensional space. If instead we recognize that long texts can be classified according to several different main topics, and contain as well a sequence of subtopical discussions, we have a new basis on which to determine in what ways long documents are similar to one another. In this paper we are focusing only on accounting for main topic information; the recognition of subtopic structure for information retrieval is a problem unto itself and beyond this paper's scope (although it is discussed in a preliminary fashion in (Hearst 1993) and (Hearst & Plaunt 1993)).

## 3 Using Multiple Category Information

We propose the use of category information as a way to give context to retrieval results from long texts. Furthermore, we emphasize the need for assignment of *multiple* categories per document, where the categories are assumed to be somewhat independent of or orthogonal to one another. Thus two topic categories that are not usually considered semantically similar can nevertheless be associated with the same text if it happens to be about both topics.

If we associate multiple main topic categories with each text, users can browse the results of initial queries with respect to these. The categories serve the dual purpose of (i) orienting the users as to the nature of the dataset (if it is unfamiliar) and (ii) helping them sort through an initial large set of returned documents in order to choose some for query reformulation and/or relevance feedback.

In order to facilitate (i), the category sets should be tailored to the text collections they are assigned to. For example, a user interested in local area networks might tap into an unfamiliar text collection. Assume that when the user queries on the word "LAN", the system returns general categories, i.e. *technology, finance, legal*, etc. If the user is interested in topics such as the impact of LAN technology on the business scene, then this dataset might be useful. If on the other hand the user wants technical information, the contextualizing information makes it clear that the search should be taken to another dataset. If the same query on a new dataset returns categories like *file servers, networks, CAD*, etc, then the user can conclude that a technical dataset has been found, and can make subsequent queries more technical in nature.

A potential problem with the use of categories to facilitate (ii), helping the user sort through the results of queries, is that it results in a need to display information with a high dimensionality. In Section 5 we introduce a browsing interface paradigm that addresses this problem. Our approach to the display of multi-dimensional information is to provide the user with a simple way to control the how much is seen at a time. The interface allows users to view the results of the query graphically, according to the intersection of assigned categories, using a Venn-diagram paradigm.

Library catalog systems have long provided categorization information in the form of subject headings. Researchers have reported that these kinds of headings often mismatch user expectations (Svenonius 1986), (Lancaster 1986). However, there is also evidence that when such subject heading information is combined with free text search, results are improved (Markey *et al.* 1982), (Henzler 1978), (Lancaster 1986). Here we are suggesting the combination of category information with term search capabilities.

The next section describes our algorithm for automated category assignment.

## 4 The Classification Algorithm

The algorithm described here is a modification of a disambiguation algorithm described in (Yarowsky 1992). The disambiguation algorithm assumes each major sense of a homograph can be assigned to a different thesaurus-like category. Therefore, an algorithm that can categorize an instance of a term according to which category it belongs to can in effect disambiguate the term. The disambiguation is accomplished by comparing the terms that fall into a wide window surrounding the target term to contexts that have been seen, in a training phase, to characterize each of the categories in which the target term is a potential member. A training phase determines which terms should be weighted highly for each category, using a mutual information-like statistic. The training does not require pre-labeled texts, rather it relies on the tendency for instances of different categories to occur in different lexical contexts to separate the senses. After the training is completed a word is assigned a sense by combining the weights of all the terms surrounding the target word and seeing which of the possible senses that word can take on has the highest weight.

In order to categorize main topics of texts, instead of choosing from the set of categories that can be assigned to a particular target word, the algorithm measures how much evidence is present for *all* categories, independently of what word occurs in the center of the context being measured. After the entire document has been processed, the categories with the most evidence are considered to be the main topic categories of the text. This algorithm is based on the assumption, discussed above, that main topics of a text are discussed throughout the length of the text.

An advantage of the scheme described here is that it uses co-occurrence information to classify terms into pre-defined, intuitively understandable classes, as opposed to classes derived from the data. Although this kind of derivation can be useful in some instances, intuitive categories are important when interfacing between the system and the user.

Another advantage of the algorithm is that it can accommodate multiple category sets. Categorization algorithms based on clustering can only present one view on the data, based on the results of the clustering algorithm, but as shown above, documents may be similar on only one out of several main topic dimensions. Algorithms that train on pre-labeled texts can also represent multiple simultaneous categories, but are confined to using only the category sets that have been pre-assigned (since in most cases thousands of pre-labeled documents are necessary to train these algorithms).

In our framework, a category is defined by the set of lexical items that comprise it. The set of 106 general categories used to characterize the AP data was derived from WordNet (Miller *et al.* 1990), a large, hand-built online repository of English lexical items organized according to several lingustic relations. The algorithm used to derive these categories is described in (Hearst & Schütze 1993), with the goal of achieving wide coverage with general categories. We used a moderate size category set in order to facilitate comparisons against judgements made by human subjects (who would be overwhelmed by too large a category set). The algorithm has also been trained on the computer science technical reports using a set of categories derived from a loose interpretation of the ACM Computing Reviews classifications.

In an evaluation against reader judgements the algorithm was found to do better than a baseline measure but not as well as the judges. When the algorithm was allowed to select the top 7 categories to match the top 5 judges' categories, it performs almost as well. The algorithm and the results of evaluation are described in more detail in (Hearst 1994).

There exist other systems in which multiple categories are assigned to documents, e.g., (Masand *et al.* 1992), (Jacobs & Rau 1990), (Hayes 1992). However, unlike the method suggested here, these systems require large volumes of pre-labeled texts in order to perform these classifications. Knowledge-based systems can be effective text classifiers, e.g., (Riloff & Lehnert 1992), (Jacobs 1993), and (Fung *et al.* 1990), but are expensive to construct for each new domain.

The approach of (Liddy & Paik 1992) is most similar to ours. It uses Subject Code assignments from the LDOCE dictionary, creating in effect a set of general categories. Heuristics are used to determine word senses based on how many words that can be assigned a particular code occur in a sentence, as well as how likely it is for the candidate codes in the sentence to co-occur. Thus it also does not require pre-labeled texts but it does require a large number of words to have been assigned to categories in advance. (Our algorithm also requires some terms to be assigned to each category in advance, but it also automatically chooses additional terms from the corpus to act as strong indicators for each category.) It would be useful to run an experiment comparing the results of the two algorithms.

# 5 The Browsing Interface

We have developed a prototype of an information retrieval system based on a new browsing paradigm. Instead of showing how similar retrieved documents are to one another or to the query, the system shows how similar the documents are to a set of (possibly) independent categories.

The interface, called Cougar, combines keyword and category information – users can search on either kind of information or both. This allows users to get a feeling for document similarity based on the main topic categories they share. Note that different documents can be grouped together as being similar based on which categories are being looked at. E.g., if one document is about the cost of removing contaminants from food and another the cost of removing contaminants from an ecological disaster, when viewed according to the *finance* category they have an intersection, whereas if the *finance* category is not selected, the two documents do not appear to have similarities.

In this particular cut on how to display information we begin with a fixed set of categories, membership in which is designed to correspond to users' intuitions. Of course this approach is flawed, both because no one set of category choices is going to fit every document set and because users will have to guess as to what categorization according to the topic really means. Nevertheless, we posit that this approach is better than requiring the user to guess why a group of long documents have been labeled as being similar to one another (and better than simply looking at a list of titles ranked by "similarity" to the query). Furthermore, since users do not have to specify in advance which categories are of interest, they are less likely to miss interesting documents just because their understanding of the classification procedure is inaccurate.

## 5.1 Cougar

Documents are assigned a fixed number of categories from a pre-determined set using our automatic categorization algorithm described in Section 4. In the current system each document is assigned its three top-scoring categories. The documents are then indexed on the category information as well as on all (non-stopword) lexical items from the title and the body. Indexing and retrieval is done using Cornell's Smart system (Salton 1971).

Two datasets have been assigned categories and indexed. The first is a subset of a collection of AP news articles taken from the TIPSTER collection (Harman 1993) (from one month of 1989) and is indexed with the general category set described in Section 4. The second is a collection of computer science technical reports, part of the CNRI CS-TR project collection, and is indexed with the computer-related categories mentioned in Section 4.

Users issue queries via the Cougar interface (see Figure 1). In the box labeled "Main Topics" there is an entry slot for category terms; the only way to place an entry in this slot is to choose a category from the listbox labeled "Categories" to the right. There is another entry slot for keywords. The user can type in terms here or select terms from another window and paste them into the entry slot. We have experimented a bit with assigning more or less weight to the category terms; currently categories and keywords are weighted equally.

After the user initiates the search (by clicking on the "Search" button), a list of titles of the top-scoring documents appears in the lefthand side of the lower portion of the interface. The number of titles displayed is a parameter that is set in Smart; in Figure 1 the top fifteen documents are shown. The top three categories for each of the fifteen documents are also retrieved and the most frequently occurring of these are displayed in a bank of color-coded buttons above a Venn diagram skeleton. The user selects up to three of the categories and sees how the documents intersect with respect to those categories. One category can be unselected in order to allow the selection of another; the display of documents in the Venn diagram changes accordingly.

More specifically, the user selects one of the categories by mouse-clicking on a category box. The system paints one of the Venn-diagram rings with the corresponding color and places document ID numbers that have been assigned this category into the part of the ring that indicates no intersection with other categories. Clicking on an ID number causes the corresponding title to be highlighted, and double-clicking brings up a window containing the document itself. The user can now unselect this category, causing the ring to become uncolored and the displayed document IDs to disappear. Alternatively, the user can choose an additional category, causing an additional ring to be painted and filled in with document IDs. If any of the retrieved documents have been assigned both of the selected categories, their ID numbers are displayed in the appropriate intersection region. Once all three rings have been assigned categories, the user must unselect one category before selecting a new one. In this way users can easily vary which subset of the category sets is active. Figure 1 shows a configuration in which all three categories have been selected.

The following examples illustrate the use of the interface in its contextualize role.

### 5.1.1 Keywords in no context.

Issuing a search on a specific term can be frustrating because it often yields no results, especially on a dataset with which the user is unfamiliar. The interface allows the user to issue a "probe" query of a more general term in order to see what kind of information turns up. For example, after querying on the word "contaminant" the eight most frequently assigned categories are *finance, meat, government, legal_system, food, weapons, trees,* and *mammals*. As the categories imply, discussions of contaminants occur in many different contexts.

For example, articles at the intersection of *government, legal_system,* and *finance* include one summarizing the Reagan administration's environmental record, another on budget considerations for Congress (including money needed to clean up nuclear contaminants) and another reporting on department of energy predictions of costs for cleaning up chemical contamination. One article labeled with *ships*, *bodies_of_water*, and *trees* (i.e., terms having to do with nature) describes the effects of an oil spill on birdlife.

Articles labeled with the *food* category include two about an incident of cyanide poisoning in yogurt. Note that if a user were interested in documents that talk about contamination in food, in order to discover this article using keywords alone, the user would have had to specify all food terms of interest. However, with appropriate category information this isn't necessary.

#### 5.1.2 Categories to Determine Relevance of Keywords

In the next example, only four of the fifteen retrieved documents in response to a query on the word "cattle" are labeled with the higher-level category that corresponds to cattle (*ungulates*). Those that are not labeled with *ungulates* are about financial matters relating to crops and foods (e.g., crop futures). Two of those that are labeled with *ungulates*, when intersected with *meat* describe cattle in the role of livestock, the third describes a cattle drive, and the fourth, whose other category labels are *countries* and *bodies_of_water*, has only a passing reference to cattle and really describes a murder related to land ownership of tropical rainforests.

By contrast, retrieving on the keyword "cow" results in articles about land disputes with Native Americans (at the intersection of *government*, *ungulates*, and *legal_system*) and grazing fees. One document that is not labeled with *ungulate* but instead with *crime*, *weapons*, and *defense*, has only a passing reference to cows and is about a robbery.

Thus the categories can be used to show whether or not a search term is actually well-represented in a text. If the text is not assigned the category that the search time is a member of, then this is a strong indicator that the term is only discussed in passing.

#### 5.1.3 Retrieving with Categories vs. Keywords

In addition to using categories to understand the results of a query, they can be used to specify the content of the kinds of information that the system retrieves. For example, when querying with the keyword terms "crime" and "education", none of the top 15 documents retrieved are classified with both of the corresponding categories. However, when querying on the categories *crime* and *education*, all fifteen of the documents retrieved are labeled with both categories. As an illustration, one document retrieved is about a boy who burned down a school so he could be transferred to his girlfriend's school.

This example is meant to be illustrative; querying via categories opens up issues that have not been discussed here but would be well worth exploring in future.

### 5.2 Variations

Other kinds of attributes besides topic assignments can be used in this manner, including attributes like author and date, and including frequent or important terms. (Although a problem with using frequent terms is that the retrieved documents may not have many most-frequent terms in common.) The important point is that the user can browse the retrieved documents according to how the attributes intersect with one another, and the user does not have to specify the attributes in advance.

### 5.3 Related Work

As mentioned above, good browsing techniques exist for the display of short documents. For example, Scatter-Gather (Cutting *et al.* 1992), (Cutting *et al.* 1993) is a query-free browsing technique that allows users to become familiar with the contents of a corpus by interactively clustering subparts of the collection to create table-of-contents-like descriptions. This technique is very effective on shorter texts but, by the arguments of Section 2, will probably be less effective on collections of longer texts. Similarly, the Bead system (Chalmers & Chitson 1992) displays documents according to their similarity in a two-dimensional rendition of multi-dimensional document space, thus not taking document structure into account, and the system of (Fowler *et al.* 1991)) displays retrieved documents in a network based on interdocument similarity.

Both VIBE (Korfhage 1991) and the InfoCrystal (Spoerri 1993) require the users to select what pieces of information the display should be organized around. The goal of VIBE is to display the contents of the entire document collection in a meaningful way. The user defines N "reference points" (which can be weighted terms or term weights) which are placed in various positions in the display, and document icons are drawn in locations that indicate the distance between the documents and all the reference points.

The InfoCrystal is a sophisticated interface which allows visualization of all possible relations among N concepts. The user specifies which N "concepts" are of interest (actually boolean keywords in the implementation, but presumably any kind of labeling information would be appropriate) and the InfoCrystal displays graphically the number of documents retrieved that have each possible subset of the N concepts in a clever extension of the Venn-diagram paradigm. When the query involves more than four terms the crystals become rather complicated, although there is a provision to build up queries hierarchically. This differs from Cougar in several respects: Cougar more easily allows a larger number of categories or concepts to be displayed and can show which documents move from one intersection of categories to another, and Cougar does not require the user to list the categories of interest in advance.

## 6 Summary

We suggest that users browsing an unfamiliar corpus should be able to issue simple queries initially, make a quick but informed judgement as to the relevance of the corpus, and only then engage in more detailed query formulation.

The results returned from the simple initial query can be overwhelming or difficult to absorb without a good mechanism for browsing. Although good ideas have been developed about how to browse the results of queries, these approaches are based on showing how similar documents are to one another, or to the query. We have argued, however, that in the case of lengthy documents, inter-document similarity is not necessarily informative enough.

We suggest that long texts be categorized according to a set of fixed categories or predefined attributes. These fixed categories serve two purposes: to orient the user as to the nature of the dataset and to contextualize the results of the query. The categories or attributes can take on a number of forms depending on what kind of information is available and/or appropriate for the corpus. In this paper we have suggested assigning categories that characterize the main topics of long texts, and have described an algorithm that can do so with some degree of success without requiring pre-labeled texts.

A consequence of allowing multiple attributes to be assigned to documents is that they make the display problem a multi-dimensional one. To handle this, we suggest a mechanism that gives the user some control over which categories are at the focus of attention at any given time, and a simple way to see how the retrieved documents are related to one another with respect to these categories.

We have developed a prototype implementation of this display paradigm; it illustrates the main points behind the ideas presented here although it could be improved with a number of small additional features, and user evaluation studies remain to be done. More importantly, in future we plan to incorporate mechanisms for querying against subtopic structure, and for allowing queries to specify subtopic terms with respect to main topic categories.

### Acknowledgments

The author would like to thank Michael Schiff, Jan Pedersen, Narciso Jaramillo, and David Hull for their helpful comments on these ideas and this paper.

## References

CHALMERS, MATTHEW, & PAUL CHITSON. 1992. Bead: Exploration in information visualization. In *Proceedings of the 15th Annual International ACM/SIGIR Conference*, 330–337, Copenhagen, Denmark.

CROFT, W. BRUCE, & RAJ DAS. 1990. Experiments with query acquisition and use in document retrieval systems. In *Proceedings of the Thirteenth International ACM/SIGIR Conference*, 349–365.

CROFT, W. BRUCE, & R. T. THOMPSON. 1987. $I^3R$: A new approach to the design of document retrieval systems. *Journal of the American Society for Information Science* 38.389–404.

CUTTING, DOUGLAS R., DAVID KARGER, & JAN PEDERSEN. 1993. Constant interaction-time scatter/gather browsing of very large document collections. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, 126–135, Pittsburgh, PA.

CUTTING, DOUGLAS R., JAN O. PEDERSEN, DAVID KARGER, & JOHN W. TUKEY. 1992. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual International ACM/SIGIR Conference*, 318–329, Copenhagen, Denmark.

DEERWESTER, SCOTT, SUSAN T. DUMAIS, GEORGE W. FURNAS, THOMAS K. LANDAUER, & RICHARD HARSHMAN. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41.391–407.

FOWLER, RICHARD H., WENDY A. L. FOWLER, & BRADLEY A. WILSON. 1991. Integrating query, thesaurus, and documents through a common visual representation. In *Proceedings of the 14th Annual International ACM/SIGIR Conference*, 142–151, Chicago.

FUNG, ROBERT M., STUART L. CRAWFORD, LEE A. APPELBAUM, & RICHARD M. TONG. 1990. An architecture for probabilistic concept-based information retrieval. In *Proceedings of the 13th International ACM/SIGIR Conference*, 455–467.

GRIFFITHS, ALAN, H. CLAIRE LUCKHURST, & PETER WILLETT. 1986. Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Science* 37.3–11.

HARMAN, DONNA. 1993. Overview of the first Text REtrieval Conference. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, 36–48, Pittsburgh, PA.

HAYES, PHILLIP J. 1992. Intelligent high-volume text processing using shallow, domain-specific techniques. In *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*, ed. by Paul S. Jacobs, 227–242. Lawrence Erlbaum Associates.

HEARST, MARTI A. 1993. Cases as structured indexes for full-length documents. In *Proceedings of the 1993 AAAI Spring Symposium on Case-based Reasoning and Information Retrieval*, Stanford,CA.

——, 1994. *Subtopic Structuring of Full-Length Documents.* University of California, Berkeley dissertation. In preparation.

——, & CHRISTIAN PLAUNT. 1993. Subtopic structuring for full-length document access. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, 59–68, Pittsburgh, PA.

——, & HINRICH SCHÜTZE. 1993. Customizing a lexicon to better suit a computational task. In *Proceedings of the SIGLEX Workshop on Acquisition of Lexical Knowledge from Text*, 55–69, Columbus, OH.

HENZLER, ROLF G. 1978. Free or controlled vocabularies: Some statistical user-oriented evaluations of biomedical information systems. *International Classification* 5.21–26.

JACOBS, PAUL. 1993. Using statistical methods to improve knowledge-based news categorization. *IEEE Expert* 8.13–23.

——, & LISA RAU. 1990. SCISOR: Extracting information from On-Line News. *Communications of the ACM* 33.88–97.

KORFHAGE, ROBERT R. 1991. To see or not to see – is that the query? In *Proceedings of the 14th Annual International ACM/SIGIR Conference*, 134–141, Chicago.

LANCASTER, F. 1986. *Vocabulary Control for Information Retrieval, Second Edition.* Arlington, VA: Information Resources.

LIDDY, ELIZABETH D., & WOOJIN PAIK. 1992. Statistically-guided word sense disambiguation. In *Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language.*

MARKEY, KAREN, PAULINE ATHERTON, & CLAUDIA NEWTON. 1982. An analysis of controlled vocabulary and free text search statements in online searches. *Online Review* 4.225–236.

MASAND, BRIJ, GORDON LINOFF, & DAVID WALTZ. 1992. Classifying news stories using memory based reasoning. In *Proceedings of SIGIR 92*, 59–65.

MILLER, GEORGE A., RICHARD BECKWITH, CHRISTIANE FELLBAUM, DEREK GROSS, & KATHERINE J. MILLER. 1990. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography* 3.235–244.

MORRIS, JANE. 1988. Lexical cohesion, the thesaurus, and the structure of text. Technical Report CSRI-219, Computer Systems Research Institute, University of Toronto.

RILOFF, ELLEN, & WENDY LEHNERT. 1992. Classifying texts using relevancy signatures. In *Proceedings of the Tenth National Conference on Artificial Intelligence*. AAAI Press/The MIT Press.

SALTON, GERARD (ed.) 1971. *The Smart Retrieval System – Experiments in Automatic Document Processing.* Englewood Cliffs, NJ: Prentice Hall.

——. 1988. *Automatic text processing : the transformation, analysis, and retrieval of information by computer.* Reading, MA: Addison-Wesley.

——, J. ALLAN, & CHRIS BUCKLEY. 1993. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, 49–58, Pittsburgh, PA.

——, & CHRIS BUCKLEY. 1992. Automatic text structuring experiments. In *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*, ed. by Paul S. Jacobs, 199–209. Lawrence Erlbaum Associates.

Spoerri, Anselm. 1993. InfoCrystal: A visual tool for information retrieval & management. In *Proceedings of Information Knowledge and Management '93*, Washington, D.C.

Svenonius, Elaine. 1986. Unanswered questions in the design of controlled vocabularies. *Journal of the American Society for Information Science* 37.331–340.

Yarowsky, David. 1992. Word sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, 454–460, Nantes, France.

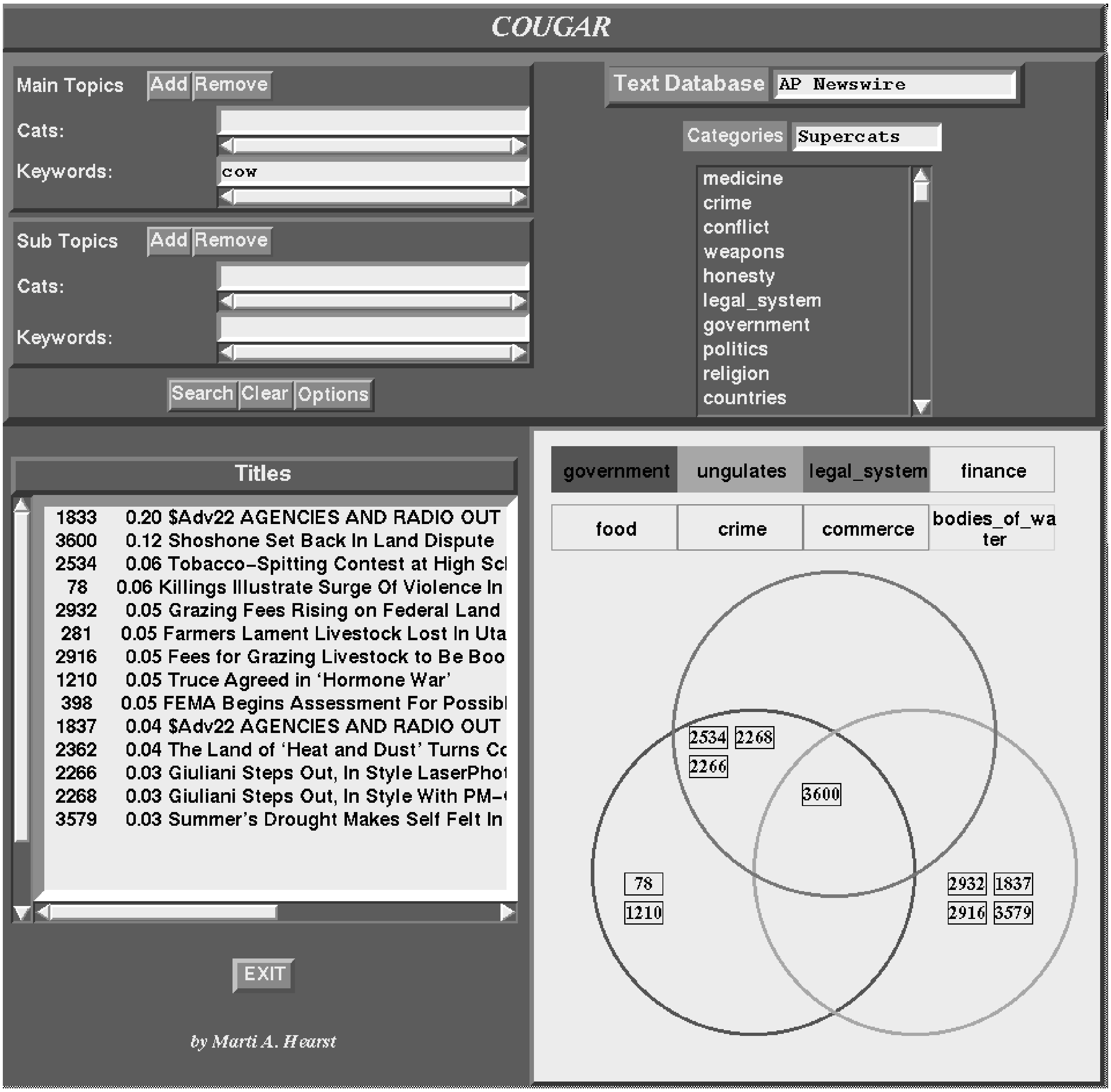



Figure 1: The Cougar interface.