[*Index*] [*Previous*] [*Next*] [*Postscript*]

CHAPTER 5

INFORMATION RETRIEVAL ISSUES

1. Introduction

Navigation or browsing is effective only for small hypertext systems. For large hypertext databases, information retrieval (IR) through queries becomes crucial. Conklin had suggested that search and query mechanisms can present information at a manageable level of complexity and detail [Conklin, 1987]. Halasz's view was that "navigational access itself is not sufficient. Effective access to information stored in a hypermedia network requires query-based access to complement navigation..........search and query needs to be elevated to a primary access mechanism on par with navigation." [Halasz, 1988].

2. Query and Search Mechanisms

Conventional IR systems focus on keyword based automatic searching (in conjunction with Boolean operations), weighting of words based on their statistical properties, ranking of documents according to probability of relevance, automatic relevance feedback for query modification and query languages [Croft et al., 1990]. However, very few (or none) of these methods retrieve complete or accurate information. Too general a query may yield a lot of items and too specific a query may retrieve no items. Thus, traditional IR is an inherently uncertain process. Combining inference techniques could eliminate or minimize uncertainty. In hypertext systems, a weighted keyword search combined with hypertext links can improve IR by finding only a subset of nodes or "hits" whose links can then be followed to other semantically related nodes [Carlson, 1989].

According to Halasz, query and search mechanisms can be classified into content search and structure search [Halasz, 1988]. Content search is standard IR technique extended to hypertext systems. That is, all nodes and links are treated independently and examined for a match to the given query. On the other hand, structure search will yield the hypertext sub-network that matches a given pattern. Query facilities which combine aspects of both content search and structure search will be capable of acting as filters. Based on the user's query, the interface will display only those nodes and links that match the query, filtering out other parts of the network. Filtered browsers have been implemented both for NoteCards and Tektronix's Neptune. In NoteCards, a user can filter out information based on the node or link type. In Neptune, the query can be content-based; if the query is broad enough, a global view of the entire network is displayed; if the query is well refined, the viewing size will be manageable.

2.1. Content Queries and Indexes

Bruza proposed a two-level architecture for hypertext documents, the top level called hyperindex (containing index information) and the bottom level hyperbase (containing content nodes and links) [Bruza, 1990]. The hyperindex consists of a set of indexes linked together. When an index term describing the required information is found, the objects from the underlying hyperbase are retrieved for examination. Navigating through the hyperindex (not the hyperbase) and retrieving information from the hyperbase is called "Query By Navigation" [Bruza, 1990].

An index is made of a set of index entries. Each index entry consists of a term descriptor or keyword and a locator (like a page number). Term descriptors lack specificity. Term phrases are made of term descriptors thus increasing specificity. However, they may retrieve too many items or no items at all and hence lack exhaustivity. Index expressions provide relationships between term descriptors. Thus, they are more specific than term phrase descriptors. Index expressions have a structure that can be used to derive a lattice of descriptors supporting query by navigation. A base index expression consists of terms that are linked to other terms by connectors. For example, "effective information retrieval" is a base index expression. So is "people in need of information". The two combined together form an index expression. For example, "effective information retrieval information AND people in need of information".

The power base expression is a lattice formed out of a full base expression at the top and an empty base expression at the bottom. This lattice (or lattice-like) structure is the basis of the hyperindex [Bruza, 1990]. Based on the vertex of focus in the lattice, the surrounding descriptors can represent enlargements (context extension) or refinements (context contraction) of the context represented by the focus. Thus, the reader can move across the lattice by refining or enlarging the current focus until a focus is found which is relevant to the information required.

Bruza’s measures to determine the effectiveness of index expressions in the hyperindex include:

a. ***Precision***: The ratio of relevant objects associated with the descriptor to the total number of objects associated with the descriptor.

b. ***Recall***: The ratio of the number of objects associated with the descriptor to the total number of relevant objects.

c. ***Exhaustivity***: The degree to which the contents of the objects are reflected in the index expressions.

d. ***Power***: The ratio of a descriptor’s specificity to its length.

e. ***Eliminability***: The ability to determine the irrelevance of a descriptor and stop the search.

f. ***Clarity***: The ability to grasp the intended meaning of the descriptor.

g. ***Predictability***: The ability to predict where relevant descriptors can be found in the index.

h. ***Collocation***: The extent to which the relevant index terms are near each other in the index.

Experiments and empirical studies are required to determine these retrieval measures for hypertext-based IR systems.

2.2. Structural Queries

Beeri and Kornatzky have suggested a logical query language that would allow structural queries over a hypertext network. First-order logic is too general since it ignores the particular characteristics of hypertext. Hence, there is a need for a structural query language to incorporate the notions of recursive and quantification constructs. The logic for the proposed query language is a mixture of propositional calculus (which has no predicates or variables) and quantifiers such as many, most, at least two, exactly five etc. [Beeri & Kornatzky, 1990]. The basic formulae of the logic are the propositions and assertions on attributes' values. Queries use specifiers to directly retrieve edges, paths, and cycles. The set of elements retrieved is collapsed into a hypertext network. The output of a query being a hypertext network, users can incrementally compose queries. Thus, the combination of specifiers, quantifiers, and the collapsing of query answers into a new hypertext network makes it possible to express structural queries proposed by Halasz. Facilities also exist in this query language to view portions of the retrieved network based on the specification of filters. Research is also underway to develop a visual hypertext query language.

GraphLog is a visual query language implemented on top of the Neptune hypertext system front-end to the Hypertext Abstract Machine (HAM) [Consens & Mendelzon, 1989]. It addresses the following issues raised by Halasz: Search and query mechanisms, augmentation of the basic node and link model, virtual structures, computation over the hypertext network, and versioning. Using GraphLog, queries are formulated by drawing graph patterns. These patterns are then searched for in the hypertext network to yield subgraphs.

GraphLog is highly expressive and it uses the notions of deductive database theory and descriptive complexity. The language is powerful enough to allow the specification and manipulation of arbitrary subsets of the network and supports the computation of aggregate functions on the subgraphs of the hypertext document. It can support dynamically defined structures as well as inference capabilities. GraphLog is an extension of the graph-based query language called G+. G+ was extended by adding the notions of negation, aggregation, and improved semantics to make it simple, yet powerful. It can express structural queries which cannot be expressed in conventional database languages such as relational algebra. For example, it can perform an arbitrarily long sequence of join operations for which there is no equivalent single relational algebra query. It was designed to avoid explicit use of logic formulae and recursion.

The concept of incremental construction of queries can also be found in HyperBase (Schutt & Streitz, 1990]. It provides two sets of retrieval operations, one set with operations that affect the whole hypertext database, another set of functions that operate only with respect to a sub-network. Streitz et al., are also working on the design of a Hypertext Query Language (HTQL).

2.3. Inference Networks

Indexes can also be considered as "precompiled links", providing immediate access to required information without navigating through the "document space" [Frisse & Cousins, 1989]. While nodes (containing information) can be considered part of the document space, indexes or index nodes can be treated as part of the "index space". In traditional full-text document retrieval systems, the document space and index space are essentially flat (they are together). Frisse and Cousins suggest the use of a hierarchical index space and a networked document space for information retrieval in hypertext systems. They have investigated the representation of index spaces as belief networks. Croft and Turtle have also proposed an IR model for hypertext systems based on plausible or non-deductive inference using Bayesian inference networks [Croft & Turtle, 1989]. Belief networks or Bayesian inference networks are directed, acyclic dependency graphs where nodes represent propositional variables and links or edges represent probabilistic relationships between the propositions. A hypertext system can be compared to such a network - the roots of the dependency graph are hypertext nodes; interior nodes and leaves represent concepts.

User's likes and dislikes are transmitted recursively from all nodes representing concepts (in the document space) to nodes in the index space. Based on this degree of belief, appropriate values are assigned to index space nodes. These changes are propagated throughout the index space using standard Bayesian techniques. If a proposition represented by a node **p** directly implies the proposition represented by node **q**, a directed graph is drawn from **p to q**. If-then rules in the Bayesian network are interpreted as conditional probabilities, that is, a rule **A --> B** is interpreted as a probability **P(B|A)**. Given a set of prior probabilities for the roots of the DAG the probability associated with the remaining nodes can be computed. Belief values of nodes of interest to a reader increase in value while nodes not of interest generally decrease in value. These probabilistic inference techniques applied to a hierarchical index space greatly enhances the information retrieval process in hypertext or networked document spaces.

Optimal retrieval effectiveness can be obtained by ranking nodes according to estimates of the probability that the query is true given a particular hypertext node. The ranking function is approximately equivalent to giving each node a score that is the sum of the weights of matching query terms, where the weights depend on the frequency of occurrence of a term in each hypertext node and in the entire collection of hypertext nodes. This simple retrieval model can be further extended by introducing dependencies that represent links between hypertext nodes. This means if hypertext node **j** is indexed by a particular concept and is linked to a hypertext node **k**, then there is some probability that hypertext node **k** should also be indexed by that concept.

However, the application of Bayesian techniques to a complex graph is NP-complete. It is also not clear whether adaptive IR systems based on Bayesian inference techniques converge on the appropriate set of index space nodes given imprecise information. Frisse and Cousins are of the opinion that the computational complexity of this approach needs more investigation. Some of the other restrictions include that the network topology cannot include directed cycles and that queries with evaluable predicates such as greater than, less than etc., cannot be handled.

Lucarella suggests a similar model for hypertext-based IR. While content nodes form the document network, there can be concept nodes forming the concept network (Figure 5.1) [Lucarella, 1990]. These concept nodes can serve as an index to the document network. The concept network is similar to the index space proposed by Frisse and Cousins. Links within the concept network establish associations between concepts. The type of the link can describe the nature of semantic association while a weight assigned to it can reflect the strength of the association. The resulting hypertext knowledge base (containing the concept network embedded within the document network) can be used for query analysis.

Based on a natural language query, the system will perform a search on the concept network to find the most pertinent sets of concepts. Concept recognition takes place by matching the various terms in the query and evaluating the matched expressions by applying similarity functions. Based on the degree of similarity between the concept and the matched item, a weight is associated to each concept. The search can be stopped based on the number of concept nodes examined and the number of top ranking concepts identified.

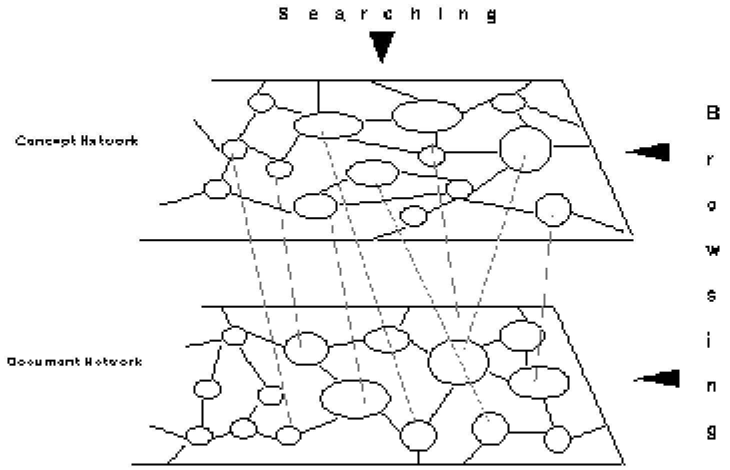

Figure 5.1. A Model for Hypertext-Based Information Retrieval [Lucarella, 1990].

Information is retrieved from the document network based on these retrieved concepts. Along with the exact set of documents, other sets of documents found to be semantically related to the topic of interest will be retrieved based on the weights assigned to the concepts. For example, a query to retrieve documents concerned with "expert systems" may also retrieve documents discussing "knowledge-based systems" since the two concepts are semantically related.

2.4. Cluster Hierarchies, Aggregates, and Exceptions

Researchers have suggested using the vector space model to organize a hypertext collection into clustered hierarchies [Crouch et al., 1989]. In this model, the content of each node or document is represented by a set of possibly weighted terms. Thus, each document can be represented by a term vector and the complete document collection can be represented by a vector space whose dimension is equal to the number of distinct terms to identify the documents in the collection. Similar or related documents are represented by similar multi-dimensional term vectors. Such a model facilitates clustering documents based on their similarity and ranking retrieved documents in decreasing order of their similarity to the query vector. Hence, the user can readily focus the search on those clusters that are likely to contain documents which are similar to the query. Comparisons are generally made between the query vector and the document vectors using one of the standard measures of similarity. Clustering is also helpful in locating neighboring nodes which discuss related topic(s). The user can incrementally refine the query vector to retrieve the desired document(s). An interactive browser incorporating the cluster hierarchy model was implemented by Crouch et al., on a Macintosh connected to a SUN network running the SMART Information Retrieval system. This interactive browser yielded a significant improvement over automatic cluster searches.

The object-oriented concept of abstraction (generalization/aggregation) will greatly benefit IR in hypertext systems [Botafogo & Shneiderman, 1991]. Abstraction is the concealment of all but relevant properties of an object or concept. Aggregation is the clustering of related objects to form a higher level object. For example, wheels, chassis, engine, body etc., can be aggregated together to form the composite object called "car". Generalization is the property of treating a set of similar objects as a generic object. For example, cars, trucks, buses etc., forms the generic object called "automobile". These two concepts of abstraction can be effectively used to simplify hypertext structures. A set of related nodes and the links between them can be treated as a semantic cluster (as opposed to the hierarchical cluster proposed by Crouch et al.) having the following properties:

a. They form a subgraph of the hypertext graph.

b. The compactness (the degree of interconnectedness of the hypertext)

of the subgraph is higher than the compactness of the whole graph.

Graph theoretical algorithms such as biconnected components and strongly connected components can be applied in the formation of such clusters or aggregates. A similar approach called Aggregation Clustering With Exceptions (ACE) has been proposed to identify aggregates or clusters and exceptions (those that do not fall into clusters) [Hara et al., 1991]. Such clustering mechanisms facilitate effective IR from hypertext systems.

3. Artificial Intelligence Techniques

A knowledge base and an inference engine built on top of the hypertext database can add "intelligence" to nodes and links. An interactive filter can be built which will consult with the user and call up the appropriate node. Since hypertext and AI have some common features (semantic dependencies between two pieces of information), research is required to merge the two in order to augment the intelligence of the user [Carlson, 1989]. Halasz reported that "Computation built into the hypermedia system is likely to be more efficient, especially when that computation involves extensive access to information in the network." [Halasz, 1988].

4. Information Retrieval and Browsing

Lucarella has written that whereas conventional IR techniques focus on "what to where" (we know what we want, but we wish to find out where in the database it is), hypertext browsers focus on "where to what" (we know where we are, but we want to know what is there) [Lucarella, 1990]. IR in a hypertext system can combine these two techniques to greatly enhance the process of finding relevant information. Hypertext browsing can supplement conventional IR by allowing users to discover retrieval cues that successively can be used for query formulation. Query facilities can supplement hypertext browsing by providing the user with a set of relevant nodes for browsing.

In a query language developed for HDM2, a query produces a list of references to items which can be units, composites, indexes, or guided tours. These can be used as a dynamic access structure (or virtual structure) from which further navigation can originate. The navigation space resulting from such a query or filter is called a hyperview. All subsequent requests (either queries or navigation commands) will be interpreted only within this restricted space [Garzotto et al., 1993].

5. Summary

While navigation or browsing is sufficient for small hypertext systems, more powerful information retrieval techniques become very important in large scale hypertext databases. Content queries can be used to retrieve the contents of nodes while structural queries can be used to retrieve subgraphs of the hypertext network that match a given pattern. Many researchers have investigated the possibilities of separating index information from contents thus forming an index space (or concept network) on top of a content space (or document network). These would not only facilitate IR but also accommodate dynamic linking and independent maintenance of the two networks.

Query languages are being extended to perform structural queries. These extensions include the notions of quantifiers, recursive operators, aggregation, and improved semantics. Research has also been carried out in the use of belief networks or Bayesian inference networks for hypertext-based IR. Some researchers have explored aggregating hypertext networks into semantic or hierarchical clusters. Very little work has been done in the area of merging Artificial Intelligence with hypertext. A combination of inference-based IR and knowledge-based hypertext could greatly facilitate browsing and searching. More research is required in the integration of querying techniques and browsing mechanisms. Experiments are required to measure the effectiveness of these IR techniques.

References

[Botafogo & Shneiderman, 1991]. Botafogo, Rodrigo and Shneiderman, Ben. Identifying Aggregates in Hypertext Structures, Proceedings of Hypertext ’91, ACM Press, 1991.

[Beeri & Kornatzky, 1990]. Beeri, Catriel and Kornatzky, Yoram. A Logical Query Language For Hypertext Systems, Proceedings of European Conference on Hypertext, ECHT ’90, 1990.

[Bruza, 1990]. Bruza, Peter D. Hyperindices: A Novel Aid For Searching in Hypermedia, Proceedings of European Conference on Hypertext, ECHT ’90, 1990.

[Carlson, 1989]. Carlson, Patricia Ann. Hypertext and Intelligent Interfaces for Text Retrieval, The Society of Text, MIT Press, 1989.

[Conklin, 1987]. Conklin, Jeff. Hypertext: An Introduction and Survey, IEEE Computer, September 1987.

[Consens & Mendelzon, 1989]. Consens, Mariano P. and Mendelzon, Alberto O. Expressing Structural Hypertext Queries in GraphLog, Proceedings of Hypertext ’89, ACM Press, 1989.

[Croft & Turtle, 1989]. Croft, W. Bruce, and Turtle, Howard. A Retrieval Model Incorporating Hypertext Links, Proceedings of Hypertext ’89, ACM Press, 1989.

[Croft et al., 1990]. Croft, W. Bruce, Belkin, Nicholas, Bruandet, Marie-France, Kuhlen, Rainer, Oren, Tim. Hypertext and Information Retrieval: What are the Fundamental Concepts, Panel Discussion, Proceedings of European Conference on Hypertext, ECHT ’90, 1990.

[Crouch et al., 1989]. Crouch, Donald B., Crouch, Carolyn J., and Andreas, Glenn. The Use of Cluster Hierarchies in Hypertext Information Retrieval, Proceedings of Hypertext ’89, ACM Press, 1989.

[Frisse & Cousins, 1989]. Frisse, Mark E. and Cousins, Steve B. Information Retrieval From Hypertext: Update on the Dynamic Medical Handbook Project, Proceedings of Hypertext ’89, ACM Press, 1989.

[Garzotto et al., 1993]. Garzotto, Franca, Mainetti, Luca, and Paolini, Paolo. Navigation Patterns in Hypermedia Data Bases, Proceedings of the Twenty-Sixth Hawaii International Conference on System Sciences, 1993.

[Halasz, 1988]. Halasz, Frank. Reflections on NoteCards : Seven Issues for the Next Generation of Hypermedia Systems, Communications of the ACM, July 1988.

[Hara et al., 1991]. Hara, Yoshinori, Keller, Arthur M., and Wiederhold, Gio. Implementing Hypertext Database Relationships through Aggregations and Exceptions, Proceedings of Hypertext ’91, ACM Press, 1991.

[Lucarella, 1990]. Lucarella, Dario. A Model For Hypertext-Based Information Retrieval, Proceedings of the European Conference on Hypertext (ECHT) ’90, 1990.

[Schutt & Streitz, 1990]. Schutt, Helge A., and Streitz, Norbert A. HyperBase: A Hypermedia Engine Based on a Relational Data Base Management System, Proceedings of ECHT ’90, 1990.