Information Agents for Building Hyperlinks

James Allan, Jim Davis, Dean Krafft, Daniela Rus, Devika Subramanian*
Department of Computer Science
Upson Hall
Cornell University
Ithaca, NY 14853
email: rus@cs.cornell.edu
FAX: (607) 255-4428

Abstract

In [13] we describe an architecture for building special-purpose agents for solving the information capture and access problem in large, multi-media data environments with uncertainty in data representation and interpretation. The key idea is to recognize the underlying structure *that exists at various levels of granularity in such environments, and to use it to build partial models (that can serve as* indices*) which encode information about content. In this paper we apply this computational paradigm to create and maintain a rich hierarchy of indices that link documents in an electronic repository of technical reports (TR) located over a wide-area network. We describe issues in the acquisition and use of these indices in answering a wide range of queries in the TR data environment.*

1 Introduction

The *Information Capture and Access* problem in its most general form is posed as follows: given an electronic data environment, capture it by acquiring partial models and access it guided by tasks specified by users. In this paper we focus on a data environment consisting of a large electronic repository of technical reports distributed over the Internet. The tasks are queries that support on-line browsing of technical reports. Queries include content-based retrieval of reports (for instance retrieving papers that demonstrate the scalability of a given distributed computing protocol, or papers on mobile robotics with pictures of mobots), as well as access of reports via citations.

This paper is organized as follows. First we review our approach to information capture and access based on information agents and explain our methodology for constructing them. Next we show how to use information agents for organizing information in the technical report repository by building complex indices of documents with various levels of information about content of the documents. Finally, we illustrate the use of these indices in building sophisticated hyperlinks across documents and give examples of classes of retrieval tasks that rely on such structures. We also argue how these hyperlinks can be used to create a good interface for the on-line reading of technical reports.

1.1 Previous Work

The proliferation of information in electronic form, the development of high-speed networking, and the evolution of digitized video and audio, makes the retrieval of relevant information from imprecise query specifications one of the grand challenges of computer science. Tools like WAIS [9], Gopher, Netfind [17], and Archie have been developed in response to this challenge. We are inspired by research in four distinct areas: information retrieval and filtering [1, 3, 14, 16], mobile robotics[4, 5], knowbots as proposed in [10], and automated document structuring[18, 6, 11].

2 Information Agents

Our approach to information capture and access [13] is to decompose the problem into recognition of high-level structure and traversal of portions of the data environment that exhibit this structure. As an example, consider the retrieval of precision recall measures for specific collections (such as the CACM) from a scanned set of technical reports on information retrieval. Using knowledge that precision recall numbers are frequently displayed in tables, we extract tables from results sections with an efficient table detector and then traverse the extracted tables to determine information about CACM. Thus in the recognition phase, we build maps of the environment that reveal high-level units of information relevant to the query. In the traversal phase, we examine these units to extract required answers.

This approach is effective in information domains with data represented at a level of detail that is too fine for exhaustive searches to be feasible (*e.g.* bits, for scanned text), but where there is enough underlying structure to filter relevant information efficiently. Our notion of structure is any level of abstraction higher than the basic unit of data representation. For example, tables, figures, lists, paragraphs and sections are layout-based abstractions for documents. Theorems, lemmas, examples and counterexamples are content-based abstractions. These structures or abstractions

*The authors are listed in alphabetical order. Address correspondence to Daniela Rus.

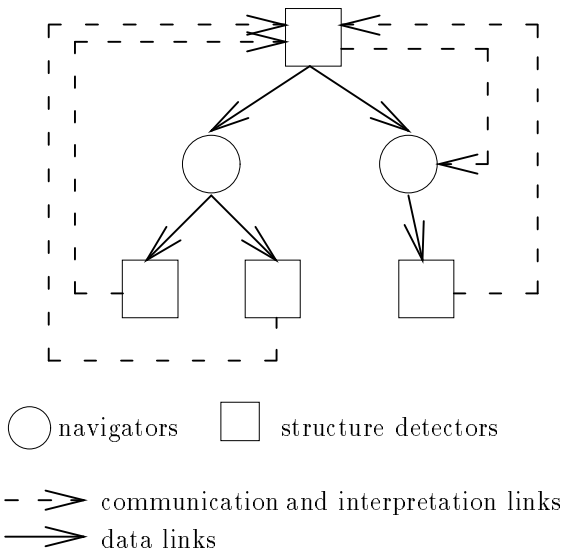

Figure 1: The architecture for Information Capture and Access. The tree alternates structure detectors an navigators.

encode high-level information about the data, and serve as filters to select the portions of the data relevant to a task. One can view structures as high-level indices into a data environment in analogy with keyword indices associated with plain text.

Our method to solve information capture and access problems uses information agents. Information agents are programs that sense and act in the information world. An information agent is a composition of special-purpose programs called *structure detectors* that handle recognition of structure and *navigators* which are responsible for traversal. Structure detectors efficiently decide whether a block of data has a specified abstract property: "is this block of text a table or a graph?" Navigators traverse the data segments extracted by structure detectors. They actively make maps at various granularities of representation. For instance, the table navigator for the stock price agent extracts models of the table as a set of rows and refines it as a set of characters in each row. This model of the data environment can be used by a character-level structure detector like the Unix *grep* utility to find the string "CACM". An information access task is solved by constructing an agent that roams the environment, aided by its navigators, looking for patterns, recognized by its structure detectors. This approach to Information Capture and Access bears a strong similarity to the search of physical environments by mobile robots. Structure detectors are virtual sensors and navigators are virtual effectors.

More specifically, an information agent is a tree of structure detectors and navigators glued together by communication paths [13], as in Figure 1. The root of the tree is a structure detector for the user's query. We view the query as a request for detecting and retrieving some specific structure from the world. The tree consists of alternating layers of structure detectors and navigators. Arcs between the nodes are communication paths. There are two types of communication paths. The first type flows downwards through the tree, connecting structure detectors and navigators. Structure detectors select partial models of the world for the navigators below them. Navigators take partitions of the world and traverse them producing topological refinements for the structure detectors below them. In addition to these data paths, an information agent also has communication and interpretation links, that can connect any two nodes in the tree. The communication and interpretation paths are used for answer assembly, and for error detection and recovery.

Information agents have been designed to improve access to information over the mechanisms provided by servers like WAIS [9] and Gopher. These systems allow a user access to very large ASCII text documents. Queries are posed by specifying keywords. When word-based indices do not exist or are inadequate, queries are either not supported, or are answered by matching a regular expression over characters by scanning every character in the collection of documents. This low-level access to data limits the class of tractable queries. With increasingly rich sources of information available over the network in the form of scanned documents, audio, and video, there is a need to enlarge the scope of queries and access pieces of documents by structural units larger than individual words. Take the query "fetch the table that summarizes the closure properties of families of languages from *Introduction to Automata Theory, Language, and Computation* by John Hopcroft and Jeffrey Ullman." A pure text-based system augmented with good indexing fails to retrieve the table. As another example, consider the retrieval request "get the paper by Bruce Donald with a photograph of Tommy[1] in it". These requests can be handled by structure detectors that can identify higher-level entities like tables and photographs. Yet another example is the query "find a mechanism that converts a uniform rotary motion into a reciprocation in Artobelovsky's design encyclopedia [2]". Paper encyclopedias are typically indexed by the component types in the mechanism, and not by behavior. To answer this query, a structure detector that embodies knowledge of kinematics is used, much like the structure detectors for the previous examples that embody knowledge of document layout.

3 Using Information Agents to Build Indices

Our methodology of building information agents relies on the fact that structure is a high-level cue for content. Electronic documents have a multiplicity of content and layout-based structures that are not exploited by traditional, keyword-oriented retrieval methods. This hidden structure can be computed, and

[1] Tommy is the name of the mobile robot designed and used in the Cornell Robotics and Vision Laboratory.

used to both speed-up and enrich the class of retrieval queries. As an example consider technical reports that are stored as a set of bitmapped files, each representing a page. The top-level index in this case consists of the physical location of the pages of the document. Another possible indexing structure reflects the division of the report into sections, rather than pages. This index allows retrieval of sections relevant to a query without examining all the scanned pages. For each section, we can construct finer-grained indices that identify section components, such as text paragraphs, pictures, tables, graphs, theorems, proofs, references, algorithms, *etc.* Thus, an index is simply a pointer to a segment of the data environment; indices are "hooks" of layout and content at various granularities into the world.

We can view the information agent approach to queries over electronic documents as the *on-line* creation of *multi-level* indices, as in automated document structuring [18, 6, 11]. In an information agent, the process of capturing and accessing information consists of a sequence of steps of structure detection and navigation. At each step, the data is partitioned into blocks by navigators. Structure detectors associated with those navigators recognize the meaning, or *type* of each block: paragraph, picture, table, theorem, *etc.* Each type serves as the basis for creating an index. To go from one step to the next, the partition of the data is refined to consist of smaller segments. The process of refining block partitioning and classification results in a hierarchy of partial models for a document. We have designed and implemented agents for detecting tables, graphs, polygons, paragraphs, itemized lists, and an agent to locate bibliographic references across the Internet.

Each partition of the data environment makes certain queries easy to answer while making others difficult or impossible to handle. In a technical report, the figures that go with a given section are often not located within the pages that contain the text of that section. In the section-oriented partition of the data, we transcend the geometric and aesthetic constraints imposed by physical page layout in the scanned technical reports, and build logical links between pages to treats sections as indivisible units. Such a conceptualization supports queries to retrieve figures associated with a section irrespective of their physical location in the document. The section-oriented partition is unsuitable for queries of the form: "retrieve the paper in the TR repository with a picture of the mobot Tommy in the right-hand corner of the second page."

Partitioning the data environment is actually described as computing topologies [13]. The bitmapped representation of the documents forms the base of an information lattice. The elements in the lattice are subsets of the base data environment. The sections of the document are subsets of the base bitmap and they form a partition of the bitmap. We can compute aggregate properties of the sections (for instance, their length) from this partition. Information agents dynamically construct the lattice given the base element, in response to user queries. The lattice can be viewed as a multi-level index revealing various views of the document at different levels of detail to facilitate fast identification of information relevant to the user.

4 Using Information Agents for Hyperlinks

Hyperlinks are semantic links between different partitions of the world. They provide a layer of structuring above the hierarchy of indices described in the previous section. Information agents are naturally suited for building hyperlinks: their navigators aid the traversal of the world via indices, and their structure detectors discover similarities. Based on the tools used for discovering similarity between various pieces of the documents, the information agents build three types of hyperlinks: layout, content, and mixed. Layout links are computed by geometric matching algorithms, like testing for the similarity of two polygons by using the Hausdorff metric [8]. Content similarity between two blocks of text can be computed by using text retrieval systems like Smart. In [1] a method for automatically generating a graphs that represents similarity between documents is presented. This graph can serve as input to graph-theoretic algorithms that select subgraphs with certain desirable properties. Mixed similarity is detected by combining layout and content techniques.

4.1 Layout Hyperlinks

The layout hyperlinks are detected by information agents whose structure detectors retrieve and recognize non-text blocks: tables, graphs, pictures, figures, *etc.* Layout similarity provides a basis for content similarity. Consider a partition of the technical reports by topic. Of all the papers on robotics, those with a picture of Tommy in them refer to theory and experiments for cooperating mobile robots. Another example of semantic linking is collecting all the papers on the ISIS system which contain performance graphs. The collection of graphs provides the user with data on the performance evolution of ISIS. Other examples include recognizing similarity between computational geometry papers containing the same kinds of polygons as hard cases and/or counterexamples, or recognizing similarity in chemistry papers based on molecular structures.

Information agents compute the layout links efficiently by starting from indices of the requested type. For instance, papers with similar polygonal structures can be found by creating an agent with the index of type "polygon" as the top-level navigator, and by using the Hausdorff distance matching algorithm [8] as top-level structure detector. Similarly, papers across the Internet that contain pictures of mobile robots can be collected by creating an agent with the index of type "photograph" as the top-level structure detector [13], and by using vision algorithms to recognize given robots in a photograph.

4.2 Content Hyperlinks

The content hyperlinks are detected by information agents whose structure detectors retrieve and recognize similarity in text blocks. The Smart system provides a tool for computing word-based similarity

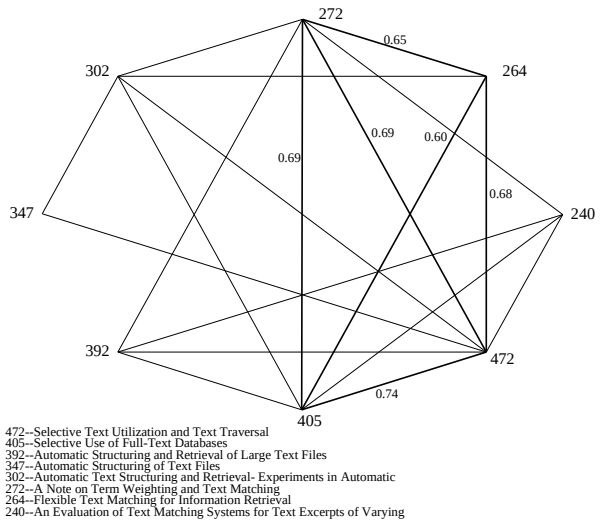

Figure 2: A similarity graph with an identified clique. The nodes of the graph represent technical reports on information retrieval from the department of computer science at Cornell University. The clique of highly similar documents in this graph is designated by thick edges. Only the weights of the edges in the clique are shown.

between documents. In [1], a method of automatically building weighted graphs that describe similarities between a collection of text blocks is proposed. Subgraphs of this graph with certain properties reflect similarity at various levels. For instance, a clique in the similarity graph calculated at the paragraph-level, where all the edges have a very high weight identifies duplication in a collection of documents on the same topic.

Information agents compute the content hyperlinks by first employing the Bib agent described in [13] to find all the papers on a given topic in the distributed repository of technical reports. This data set is filtered by a Smart structure detector [1] whose results (returned in the form of a graph) are further processed by a structure detector that recognizes interesting subgraphs. Examples of such structure detectors are recognition of maximum-weight cliques as in Figure 2 (this corresponds to duplicate paragraphs) or recognition of trees of height 2, where the root is connected directly to leaves (here, the root represents a paragraph that is expanded in the section consisting of the leaves of the tree.)

Informations agents that operate as described above use as first-level navigators the index for the paragraph-level representation of documents. Searching can be improved by using as first-level navigator the index for the section-level representation of documents. When section titles are meaningful (*i.e.* not for titles such as Introduction, Main Result, Conclusions), using Smart at the section-title level can be used to filter the most relevant sections.

4.3 Mixed Hyperlinks

The mixed hyperlinks are detected by combining layout and content knowledge. Examples include connecting a citation to the full reference and then to actual text of the reference; a theorem to its text and the text of its proof, regardless of whether or not they are all in the same document.

To construct an agent that fetches the text of the reference from a citation, we need structure detectors that have knowledge of citation styles (plain, alphabetical, APA) and can gather the actual reference from the Reference or Bibliography section of the paper in which the citation occurs. A section-level navigator constructs the appropriate partition for these structure detectors. We can compose this agent with the Bib agent in [13] that retrieves papers from the Internet given a full reference for them. The same ideas can be used to link a theorem reference to its statement and to its proof.

More challenging examples of mixed hyperlink construction include linking figures with detailed explanations associated with them in the text to handle queries that seek elaboration on the figures, as well as linking related algorithms by merging syntactic knowledge about layout of algorithms (presence of similar while loops) with semantic knowledge (e.g., input-output specifications). This allows support of queries of the form: retrieve all two-phase distributed database commit protocols that rely on a given Network failure model.

5 Summary

We advocate modular construction kits for information agents with libraries of structure detectors and navigators. They implicitly define an information lattice over the data environment providing views at different levels of detail. Information agents traverse this lattice dynamically guided by user queries. Additional layers of structure to support a richer range of queries are built using layout-based, content-based and mixed hyperlinks that reveal connections between units in the data environment.

Acknowledgements

We are grateful to John Hopcroft for his guidance and support. Thanks also to T.V. Raman, Matthew Scott, Amit Singhal, Kristen Summers, and Rich Zippel for enlightening discussions.

The work has been supported by the Advanced Research Projects Agency of the Defense Department under ONR Contract N00014-92-J-1989, by ONR Contract N00014-92-J-39, by NSF Contract IRI-9006137, and by NSF Contract IRI-8902721.

This work also been supported by the Advanced Research Projects Agency under Grant No. MDA972-92-J-1029 with the Corporation for National Research Initiatives (CNRI). Its content does not necessarily reflect the position or the policy of the Government or CNRI, and no official endorsement should be inferred.

References

[1] J. Allan and G. Salton, The identification of text relations using automatic hypertext linking, submitted to *the Workshop on Intelligent Hypertext, the ACM Conference on Information Knowledge Management*, November 1993.

[2] I. Artobolevsky, *Mechanisms in Modern Engineering Design, vols. 1-4*, MIR Publishers, Moscow, 1979.

[3] N.J. Belkin and W.B. Croft, Information filtering and information retrieval: two sides of the same coin?, *Communications of the ACM*, vol. 35(12), pp. 29-38, 1992.

[4] R. Brooks, Elephants don't play chess, *Design of Autonomous Agents*, ed. P. Maes, MIT/Elsevier, 1990.

[5] B. Donald, Information Invariants in Robotics, to appear, *Artificial Intelligence*.

[6] H. Fujisawa, Y. Nakano, and K. Kurino, Segmentation methods for character recognition: from segmentation to document structure analysis, *Proceedings of the IEEE*, vol. 80, no. 7, 1992.

[7] J. Hopcroft and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[8] D. Huttenlocher, G. Klanderman, and W. Rucklidge, Comparing images using the Hausdorff distance, to appear, *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[9] B. Kahle, Overview of Wide Area Information Servers, WAIS on-line documentation, 1991.

[10] R. Kahn and V. Clef, *The World of Knowbots*, report to the Corporation for National Researc Initiative, Arlington, VA 1988.

[11] M. Mizuno, Y. Tsuji, T. Tanaka, H. Tanaka, M. Iwashita, and T. Temma, Document recognition system with layout structure generator, *NEC Research and Development*, vol. 32, no. 3, 1991.

[12] S. E. Robertson, The methodology of information retrieval experiment, *Information Retrieval Experiment*, in K. Sparck Jones, Ed., pp 9-31, Butterworths, 1981.

[13] D. Rus and D. Subramanian, Multi-media RISSC Informatics: Retrieving Information with Simple Structural Components, in *Proceedings of the ACM Conference on Information Knowledge Management*, Nov. 1993.

[14] G.Salton and M.J. McGill, *Introduction to Modern Information Retrieval*, MacGraw-Hill, New York, 1983.

[15] G.Salton and C. Buckley, Improving retrieval performance by relevance feedback, *Journal of American Society for Information Science*, vol. 41(4), pp. 288-297, 1990.

[16] G. Salton, *Automatic Text Processing: the transformation, analysis, and retrieval of information by computer*, Addison-Wesley, 1989.

[17] M. Schwartz and P. Tsirigotis Experience with a Semantically Cognizant Internet White Pages Directory Tool, *Journal of Internetworking Research and Experience*, March 1991.

[18] Y. Tanosaki, K. Suzuki, K. Kikuchi, and M. Kurihara, A logical structure analysis system for documents, *Proceedings of the second international symposium on interoperable information systems*, 1988.

[19] S. Tsujimoto and H. Asada, Major components of a complete text reading system, in *Proceedings of the IEEE*, vol. 80, no. 7, 1992.