Exhibit 16

Information Processing & Management Vol. 23, No. 4, pp. 341–366, 1987
Printed in Great Britain.

0306-4573/87 $3.00 + .00

DEVELOPMENT OF THE CODER SYSTEM: A TESTBED FOR ARTIFICIAL INTELLIGENCE METHODS IN INFORMATION RETRIEVAL

EDWARD A. FOX
Department of Computer Science, Virginia Tech, Blacksburg, VA 24061

(*Received* 25 *February* 1987)

Abstract—The CODER (COmposite Document Expert/Extended/Effective Retrieval) system is a testbed for investigating the application of artificial intelligence methods to increase the effectiveness of information retrieval systems. Particular attention is being given to analysis and representation of heterogeneous documents, such as electronic mail digests or messages, which vary widely in style, length, topic, and structure. Since handling passages of various types in these collections is difficult even for experimental systems like SMART, it is necessary to turn to other techniques being explored by information retrieval and artificial intelligence researchers. The CODER system architecture involves communities of experts around active blackboards, accessing knowledge bases that describe users, documents, and lexical items of various types. The initial lexical knowledge base construction work is now complete, and experts for search and time/date handling can perform a variety of processing tasks. User information and queries are being gathered, and a simple distributed skeletal system is operational. It appears that a number of artificial intelligence techniques are needed to best handle such common but complex document analysis and retrieval tasks.

1. INTRODUCTION

Online searching of bibliographic data bases, originally an aid to research in areas like medicine and chemistry, has become increasingly important in serving the diverse needs of both the public and private sectors. With the spread of word processing and electronic publishing, a greater proportion of written materials now being produced is available in machine-readable form. The advent of full-text data bases [1], originally important primarily for legal research, has helped push the total number of publicly accessible online data bases above the 3,000 mark. If one includes corporate and private text collections, such as can develop from office information systems with a text filing capability, tens of thousands of searchable collections already exist.

1.1. End user searching

Machine-aided searching of early data bases began during the 1950s but was a cumbersome and expensive process. Today, with far-reaching networks connected to mainframe computer systems that manage vast banks of online storage or with powerful microcomputers controlling high-capacity CD ROM (compact disc read only memory) optical drives, individuals have the hardware tools to perform their own searches, but still only have fairly primitive software support.

Even in the domain of bibliographic retrieval, many end users prefer to locate interesting items without having to involve a search intermediary [2]. Not every user feels this need, but many search intermediaries also desire a simpler means of access to the multiple systems and data bases involved. Furthermore, in the context of office automation, individual office workers usually *must* search on their own.

Project funded in part by grants from the National Science Foundation (IST-8418877) and the Virginia Center for Information Technology (INF-85-016) and aided by an AT&T Equipment Donation.

In 1981 it was argued that searching methods should be improved and that additional research was of value [3]. Although many desirable qualities of a good interface had been identified and some experimental systems had been developed, access to information was difficult for the average user [4]. Since that time, knowledge of underlying retrieval principles has progressed to the point where textbooks applying those concepts to conventional online searching are available [5]. A few systems like PaperChase, which serves medical practitioners [6], are simple, well engineered, and powerful. Yet although there are now commercially available systems aimed at facilitating searching, and published lists of suggested techniques [7], there is still no truly helpful search software that can effectively meet the needs of most end users [8].

1.2. Need for improvement in information retrieval systems

One recent study indicated that only about 20% of the relevant items were found by careful searchers who used a commercially available full-text retrieval system [9]. Earlier work had shown the overlap between search results of different people to be small [10]. Putting these results together, one is not surprised that the effectiveness and efficiency of searching by a single individual seems to be rather low [11]. Since it is typically not possible to follow the obvious suggestion derived from these studies, namely to have several different searchers work on the same interest statement and then pool their results, it seems appropriate to consider a different approach, such as making a computer play the role of several (intelligent) searchers. In view of the advances made in automatic text retrieval systems [12], this should be feasible in the near future.

The opportunity of discovering how to dramatically improve retrieval effectiveness has challenged the body of researchers working on automatic indexing and retrieval systems and has led to the development of a variety of methods based primarily on statistical and probabilistic processing of text collections and user queries. While the marriage of these methods to powerful microcomputers and optical stores is now possible and would benefit many users [13], some researchers feel that more "intelligent" approaches are needed to provide even greater effectiveness [14]. Since artificial intelligence (AI) methods have begun to provide assistance in a variety of other complex tasks, the information retrieval (IR) problem is being re-formulated as one involving "knowledge" bases [15].

1.3. CODER and Prolog

The CODER project was proposed as a means of systematically investigating the use of knowledge-based approaches, to be implemented in Prolog, for handling the complex task of analysis and retrieval of composite documents [16]. Earlier Pollitt had suggested that logic programming methods in general, and Prolog in particular, should be applied to information storage and retrieval problems [17]. Prolog can be easily learned [18] and, although classified as an AI language, can be applied to a variety of problems [19]. Advanced texts on Prolog programming are now available, so the elegance of logic programming and the efficiency and metamathematical expressiveness of Prolog can both be properly unleashed [20]. Data base researchers find it appealing to bring together the flexibility and expressive power of Prolog and the efficiency of data base management systems [21]. A version of Prolog with a built-in data base capability, MU-Prolog [22], was therefore selected for use in CODER. Pattern matching (by unification), recursion, automatic backtracking, searching through sets of facts or rules, and list manipulation are additional features of Prolog that allow it to be easily adapted to handle lists of keywords, perform natural language parsing of queries, manage a relational data base describing document features, and make inferences about complex knowledge structures associated with the content of a text.

With such a powerful tool, it was necessary to carefully define the system requirements and problems to be addressed by CODER and to study the various related efforts under way in the IR and AI areas. These subjects are discussed in Sections 2 and 3, respectively. Section 4 follows, providing an explanation of the particular approach taken. Implementation details are then given in Section 5, and conclusions appear in Section 6.