Wide-area information access to multimedia historical sources

Tim Mills and Ken Moody
{Timothy.Mills,Ken.Moody}@cl.cam.ac.uk
University of Cambridge Computer Laboratory
Pembroke Street, Cambridge CB2 3QG, United Kingdom

Abstract

An object-oriented model has been developed for heterogeneous multimedia data; this model underlies Cobra, a content-based retrieval architecture, which allows quick construction of powerful tools for wide area information access. The system is to be evaluated through case studies, the first of which is a search engine for historical records.

1 Research Goals

The goals of this research are twofold. An object-oriented model has been developed for heterogeneous multimedia data; this model underlies an information retrieval (IR) system which allows quick construction of powerful tools for wide area information access. While current research on information retrieval is generating new methods of handling multimedia sources, systems which exploit this technology tend to be bespoke designs. This work presents research into authoring support for multimedia information retrieval systems and applications.

In order to evaluate the model and its associated retrieval architecture a number of case studies are to be undertaken. In this paper we present the 'base-case' for the architecture, namely that of text retrieval.

An important collection of historical records is being mounted on a World Wide Web server, and a search engine based on our IR model has been constructed to guide researchers in their use of the sources. These records have already been made available on microfiche, and historians from several continents have published studies based on them. By comparing earlier experiences with those of historians who are currently accessing the same data through the Web, we hope that the practical benefits of our IR system will become evident. Once we feel confident that the basic IR architecture is sound we shall move on to the next phase of the project, which is to set up a second set of documents which include photographs, video and audio components as well as straight text.

Since we regard critical evaluation by end users as central to the success of the IR project an important secondary goal of our work is to facilitate academic research based on the historical records. The documents are provided to a worldwide readership, and so registration and subsequent logging of users may allow conclusions to be drawn about the effectiveness of the system in terms of retrieval performance and usability.

2 Related Work

Work at Xerox PARC presented an object-oriented architecture for text retrieval [CPP91] which aimed to facilitate functional modularisation, mix-and-match combination of module implementations and definition of inter-module protocols.

This architecture takes a tool-centred view of the indexing process in which token streams which represent documents are passed between tools, such as stopword filters and stemmers, in a pipeline.

ECLAIR [HW92] is an extensible object-oriented class library for information retrieval. The system is implemented in C++ together with an ODMS for persistent storage. As such, this represents an application development platform for IR systems and applications.

In contrast to the Xerox architecture, ECLAIR is a document-centred design. This has the advantage that a class representing a document type can encapsulate all the specifics for handling that type. However, the design does not treat tools as separate objects, but rather as methods on the documents, and thus tools become an integral part of the document. While this was suitable for the small number of tools considered in this work, in the context of heterogeneous document types many of these methods would be redundant.

FIRE [SF95] is a design for a reusable IR framework. The framework is a program skeleton that defines the basic concepts of the IR domain and how its components interrelate. The developers aim eventually to support integration of the application's data model into the FIRE object model and to provide facilities for dealing with a number of media types including text and audio.

FIRE appears to be a complex architecture. While it is evident that single medium systems may be designed within the framework, it has yet to tackle the problems of heterogeneous collections.

3 Background to the Historical Data

In 1972 the Social Science Research Council initiated a project in which three communities would be examined with the aim of reconstructing as fully as possible the lives of individuals who lived in these selected areas [Mac89]. One of the places chosen was Earls Colne in Essex, England. The rich collection of historical records describing the village between 1400 and 1750 were transcribed, numbered and manually annotated, identifying each instance of an entity from a number of predefined categories [MHJ77]. These annotations included delimitation of dates, names, kinship relations, places and land transfers. Of particular importance to the users of these records are instances of people, places and dates.

The resulting transcriptions and indices derived from them were published in microfiche form. The plain text of the records totals around thirty megabytes, while the manually produced indexing information adds several megabytes more. Although in contemporary information retrieval terms this is not a large amount of data, for those who wish to study the material in detail the collection is of an intimidating size.

These records have been the subject of a great deal of historical research, but the amount of information and the microfiche format make navigation cumbersome. Hypertext systems provide a much more convenient way of navigating, since links can be used to allow traversal of the documents by following references to individuals, land holdings and chronology.

Free text retrieval is an important tool for finding related records. This extends the functionality already available via the microfiche publication, and enhances the usefulness of the document collection. By integrating a search engine with hypertext both browsing and searching of the collection become possible. It is therefore easy for users with different levels of sophistication to select their own way of working.

The data collection will grow as a result of research on the documents. For example, a number of PhD theses are currently in progress, and references to them will be added as they appear. Further, in future work we hope to use room abstractions to provide an environment for group discussions. This means that a web site is a more appropriate method for presenting the collection than a static database such as a CD-ROM.

4 Construction of the Hypertext

The original form of the data was inconvenient for processing, since it contained duplicate information and the manual indexing was separated from the main text.

Figure 1 illustrates the sources which were to be integrated. This process resulted in a single file for each of the historical sources from which other formats could easily be generated. An example record is given in figure 2.

The data itself presents some interesting problems. Firstly, some of the original documents were damaged or partially illegible, so certain words are ambiguous. A great deal of work has been carried out by experts

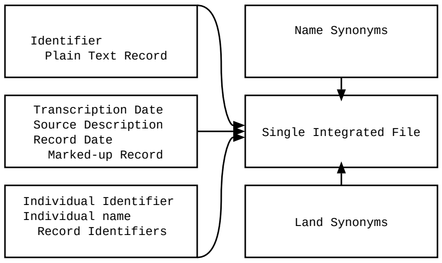

Figure 1: Information Sources

```
<DOCUMENT>
<IDENTIFIER> 0140.00081 </IDENTIFIER>
<TYPE> PRO ASSI35/18/43 </TYPE>
<TIME>  </TIME>
<FILE> ass559 </FILE>
<NOTE> 186 </NOTE>
<DATE date=19.3.1576> 19.3.1576 </DATE>
<TEXT> <NAME identity=P408 name="Giles Pawle"> Giles Pawle </NAME> of EC brickla
yer <DATE date=4.2.1577> 4.2.18Eliz1 </DATE> about six oclock in the morning the
re stole five ewes worth 25s belonging to <NAME identity=XG335 name="Wm Goody">
Wm Goodwyn </NAME> of Halstead pleads not guilty true bill <COMMENT text=" no ca
lendar of prisoners "> </TEXT>
</DOCUMENT>
```

Figure 2: File Markup Format

to determine the set of individuals referred to in the collection. Two distinct individuals may share the same name, and one individual may have more than one name, or a name may have a number of possible spellings. The situation is similar for land names. Manual indexing of these entities has meant that this domain specific information could be encoded in the file format and exploited in the hypertext and index generation process. Thus within the text, people's names are marked up with a unique identifier and a normal form of their name. Since land boundaries change over time, land names are identified by a series of numbers which refer to land regions, each identifier having a lifetime in much the same way as a reference to a person. Dates are annotated by adding a standard form, since in the original text dates could be given as day-month-year triples, month-year pairs, or single years. As a further complication, dates could refer to festivals such as Christmas, and years could be given as regnal dates, for example, 3Eliz1 refers to the third year of Elizabeth I's reign. Date ranges were also used. All dates were therefore automatically annotated with a normal form of the date.

From this unified format, it was relatively simple to generate HTML documents automatically. The markup information was used to generate hypertext links to aid navigation. Since individuals, lands and dates had been identified, indices of these attributes were generated. For example, an alphabetic index to the individuals allows access to all records containing that individual.

The generated hypertext offers a number of advantages over the microfiche publication. Both media types allow sequential navigation of the document collection, but the hypertext also allows easy movement through the collection by date, by individuals and by lands. Although this was possible with the microfiche format, the process involved finding a name, locating the identifier for that name, and then searching the name index to find record identifiers for the required records. This tedious and time-consuming sequence is replaced by a couple of hypertext links. For example, when viewing a record, a reference to an individual forms a link to the people index, thus allowing all other records referring to that individual to be reached. The hypertext indices also aid users unfamiliar with the material, since it becomes much easier to trace paths of interest via the links. This should prove particularly useful when making the collection available for such purposes as high school history projects.

The text of the documents corresponds exactly to the original, apart from translation from mediaeval Latin where needed, and some standardisation of spelling, particularly of proper names. The data is often hard to interpret, and so introductory information has been included in the site which gives a background to the collection and the period. Notes are provided to guide browsers, stressing particularly the difficulty of assigning names reliably.

Images of the selected source documents and photographs of Earls Colne provide a visual aspect to the site, including pictures of the several houses in Earls Colne which have changed little since the 17th century. Finally, maps illustrating the land plots have been coupled with client-side image maps. This allows navigation of the land index by selecting plots of land on the map.

5 A Content-based Retrieval Architecture

Cobra (*CO*ntent-*B*ased *R*etrieval *A*rchitecture is a framework for developing multimedia IR applications. The system allows construction of IR applications at several levels of complexity. At the simplest level, an application is composed of high level management modules to deal with filesystems, existing document types, user interfaces, query languages and retrieval models. The document types required for the application are registered with a type registry.

The basis of the architecture is a model for multimedia documents which combines different indexing features to build up a composite document representation. Tools such as stemming and filtering may be applied to transform between representations. Documents are described by document classes which declare the representations and tools used to represent the document, and provide methods to perform tasks such as indexing and storage.

During the indexing process, the filesystem module is used to select files for indexing, and a new document object of the appropriate type is generated by the type registry. The document object provides methods to parse the document to extract indexing features and to store the resulting representations.

Querying requires input via the user interface module. This is then parsed by the query language module which generates indexing features which are to be matched. These features must be interpreted in the context of each document type so that a single query can be matched against several document types. The query is then evaluated according to the chosen retrieval model, and the results are handed back to the user interface module for display. A further module is under development to support the relevance feedback process.

Object orientation facilitates quick exchange of component modules, and thus promotes reusability and tailorability. Through object specialisation, the system may be expanded and enhanced to meet the emerging requirements of new media types.

More sophisticated systems will depend on handling a variety of media types and formats. This is catered for by defining new document class objects. These objects define the tools and features that are used to represent the document together with a method for parsing the document. This may simply require composing existing tools and feature representations, or it may even be necessary that new tools and features be created. Of course, once written these tools and representations are available for future use.

To reduce resource usage, a tool registry manages tools so that if a request is made by two document classes for an identical tool, then the tool instance can be shared.

An efficient mechanism has also been developed for sharing representation objects: this allows the automatic creation of query classes which range over a number of document types. By providing a meta-level description of tools and representations, and allowing the order of application of tools to be expressed in the document class definition, it is possible to compare document class components. When identical components are found they can be shared, thus eliminating unnecessary duplication of work at query time.

Tool and representation sharing together achieve the modularity of document-centred approaches, since each document class can be viewed as a self-contained object defining all that is required for the indexing and retrieval of documents of that class. At the same time, the flexibility of the tool-based approaches is possible, in which tools may be added and removed to meet the particular needs of an application.

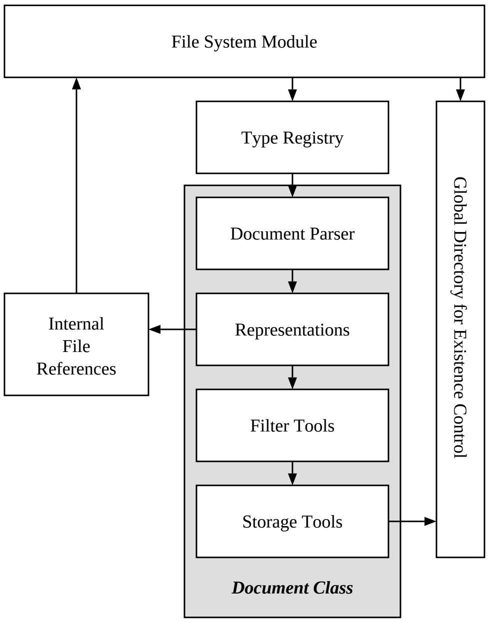

Figure 3: Module dependancies during indexing

6 The Cobra Implementation

The Cobra framework was implemented after studying the possible interactions between system components. Since modularisation aims to produce self-contained units, the points of separation have to be balanced against the functionality desired. For example, components such as the user interface may be extremely complex and require an understanding of the underlying document components, in order to provide facilities such as highlighting of search keywords in the retrieved documents. So a tradeoff between generality and reusability is inevitable. By specialising components, the functionality provided by more general modules can be reused when refining them for a particular application. Figures 3 and 4 present an overview of the interactions between components in the indexing and retrieval processes.

The current implementation supports text and image media types. The facilities for constructing new document classes currently include textual and numerical representations, dates, RGB colour values and file information. These representations may be modified or stored by tools including a morphological stemmer, stop-word filter, inverted file store, R-tree file storage and several directory services.

System components include the following modules.

User interfaces. Command line, terminal and World Wide Web form interfaces are supported.

Query language. Query languages for expressing keyword searches and date ranges have been implemented.

Retrieval models. Boolean and inference models are available.

Registries. These form the mechanism for sharing tools and representations.

As the architecture developed, it became evident that an extra facility was needed to allow for events requiring interaction with or communication to the user. So that the user can form a coherent mental model of an IR system, it is helpful to provide some indication of how a query has resulted in the retrieved documents. An example is the case of stopword filtering, in which the act of removing a common term

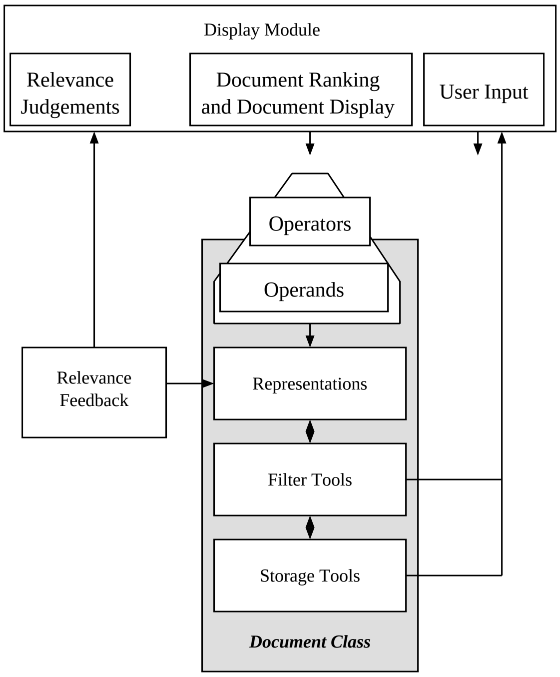

Figure 4: Module dependancies during retrieval

should be signalled to the user. Callbacks may be registered with tools so that if an event of a particular class occurs, information characterising it may be notified by the tool to the user interface. This mechanism was to prove invaluable in handling the peculiarities of the Earls Colne data.

Already a number of simple retrieval applications have been built, including a general document store, which has been used for electronic mail messages, bibliography files, manual pages etc., and a generic web site indexer.

7 Construction of the Earls Colne Search Engine

As a first step towards evaluating Cobra a searching facility has been added to the World Wide Web server for the Earls Colne data. This greatly improves the ease with which related records from the collection can be located.

The search engine was constructed to take account of the various types of information represented in the document collection by composing standard Cobra components to define a new document class. Text representations handled free text, individuals and plots of land while a date representation dealt with dates. A field representation managed the document header structure, which included fields such as the source record catalogue identifier. The tools which are applied to these representations include a stemmer and stop-word filter for the free text, and a synonym tool for the name and land representations.

The textual information is managed by dynamic inverted files [CP93], whereas the dates are indexed using a two-dimensional R-Tree [BKSS90] so that range searches may be performed. A directory module deals with URLs.

All this was embodied in the document class together with a document parser, which extracts the content-bearing features from the annotated document collection. The filesystem module handles files submitted to the indexer, and instantiates the appropriate document class for the file. Indexing is initiated by instantiating a document class for each of the document records in turn, and then issuing an index operation.

Since queries can be specified on text, names, land and dates, the user interface presents separate fields to distinguish this information. The search fields are each evaluated and the results combined to give a final result. The search engine is built from a user interface, a query language and a retrieval model. The resulting search engine is a probabilistic system based on the inference model [HW91].

Since users would find it unacceptable to refer to individuals by identifiers rather than by name, the user interface is responsible for refining the query so that the assignment of names is made apparent to the user. During the parsing of a query, if the synonym tool finds an ambiguous name, the callback mechanism is used to terminate the query parsing and to present the possible options for the user to select. This is important, since the problem of deciding how to map names to individuals has no self-evident solution. Thus, if a user chooses to disbelieve the assignments of identifiers to individuals made by the (human) indexers, they may make a selection to reflect this when refining their query.

The World Wide Web, while providing a useful interface for navigating through information and a reasonably standard protocol, is not well suited to session-based searching such as is required during relevance feedback. However, now that client-side scripting systems such as Java are becoming available, we are planning more satisfactory interfaces. The modularity of the Cobra system allows the user interface to be replaced easily. Currently command line, terminal and Web interfaces are supported.

8 Gaining experience from Earls Colne

The first application of the Cobra system has been to provide access to the text of historical records in conjunction with a hypertext, and to allow selection of the records associated with named entities such as individuals and plots of land. Currently the records are being further processed to allow access to kinship relations and to view land transactions.

We hope to augment the database with additional information such as an active map showing the changing pattern of land tenure through time. Ralph Josselin, a parish priest who lived and farmed in the village, kept an often detailed diary from 1644 until his death in 1673. The text of this diary is at present being entered into the computer, and we intend to mark it up for inclusion in the Web server.

These additions together with the background and photographic material should make the server attractive to a wider class of client, perhaps including school history departments.

More investigation is required to determine how the information can best be displayed, also the possible methods of interaction so that retrieval tools can take account of feedback. The approach being investigated at present will exploit Java. We intend to monitor sessions, and to set up a registration system so that we can record the suggestions of those who have used the server. We think that the data will have wide appeal, so that the experience we gain will be derived from users of greatly varying levels of sophistication.

9 Future Work

Mounting the Earls Colne data on a Web server will be of real service to historians interested in that period, as well as offering useful primary source material for school projects. But our main purpose is to gain understanding of the needs of social scientists before trying to evaluate the Cobra architecture in more challenging contexts. The site is nearing completion, and will shortly be made available to a small user community before large-scale access is permitted.

The Cobra system is currently being extended to handle more media types, and more representations of the supported media types. The potential of the representation model is being examined by case studies of existing real-world multimedia retrieval problems. Designed to be a practical retrieval tool, the emphasis of forthcoming experiments will continue to be on investigation of real-world problems.

In future work, the system is to be applied to multimedia anthropological sources. This information includes film clips and photographic data, and will pose extra challenges which arise from the need to retrieve documents of media types other than text. We believe that user interaction will play a key role in building a useful IR engine, and we are anxious to develop a robust system structure with text data before carrying out experiments with other media. The two sets of data have been collected by the same social historian, Alan Macfarlane, and the common underlying philosophy should help us to extend experience gained with the text retrieval system into a multimedia context.

When handling heterogeneous collections, data fusion techniques will be necessary to manage the merging of multiple sources of evidence for a document's relevance to a query. This area must be thoroughly investigated to ensure that such fusion is valid and that appropriate fusion methods are selected. A further issue is that of user interface design for multi-modal queries, which has yet to be addressed.

The real tests of the Cobra architecture will be twofold. First, will it be possible to experiment with the structure of the IR engine in a modular fashion? An example would be to modify the way in which relevance feedback causes query refinement. Secondly, can such experiments be carried out as easily with multimedia as with text documents? The object-oriented structure of Cobra and the flexible nature of the document representation have been designed with such requirements in mind, but there will be many lessons to be learnt from the experiences of real users.

Acknowledgements

We acknowledge the UK EPSRC for supporting this work through a studentship and under grant GR/J42007. We are grateful to ICL for general support of our research group.

References

[BKSS90] N Beckmann, H P Kriegel, R Schneider, and B Seeger. The R* -Tree: An Efficient and Robust Access Method for Points and Rectangles. In *Proceedings of the ACM SIGMOD Conference*, New Jersey, USA, 1990.

[CP93] D. R. Cutting and J. O. Pederson. Optimizations for dynamic inverted index maintenance. In *Proceedings of SIGIR'90*, June 1993.

[CPP91] D. R. Cutting, J. O. Pederson, and Halvorson P. An object-oriented archiecture for text retrieval. In *Proceedings of RIAO'91*, 1991.

[HW91] Turtle H. and Croft W.B. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187 – 222, 1991.

[HW92] D. J. Harper and A. D. M. Walker. An extensible class library for information retrieval. *The Computer Journal*, 35(3):256 – 267, 1992.

[Mac89] A. Macfarlane, editor. *The Origins of English Individualism*. Basil Blackwell, 1989.

[MHJ77] A. Macfarlane, S. Harrison, and C. Jardine, editors. *Reconstructing Historical Communities*. Cambridge University Press, 1977.

[SF95] G. Sonnenberger and H. Frie. Design of a reusable IR framework. In *Proceedings of the 19th ACM SIGIR*, pages 48 – 57, 1995.