A QUERY LANGUAGE FOR INFORMATION RETRIEVAL
(LANGAGE D'INTERROGATION D'UN SYSTEME DE RECOUVREMENT)

Ian A. Macleod
Department of Computing Science, Queen's University
Kingston, Ontario.

ABSTRACT

This paper outlines the query language and functional capability of the Mistral retrieval system. This is an interactive system incorporating such features as weighted keyword indexes, thesauri and user feedback. The paper is also intended to illustrate the application of an interface building system for query and command languages.
(Cet article passe en revue le langage d'interrogation et la capacité fonctionelle du système de recouvrement Mistral. Il s'agit d'un système interactif comprenant des éléments tels que des index de mots-clé pondérés, des thésauri, et de la rétroaction fournie à l'utilisateur. L'article vise également à illustrer l'application d'un système qui permet de structurer l'interface des langages d'interrogation et de commande.)

INTRODUCTION

One of the aims of this work is to develop an effective retrieval system based on automatic indexing by taking the usual inverted type of file organization, upon which most traditional retrieval systems have been based, and enhancing it with features such as a weighted index, a stopword or negative dictionary, a thesaurus and user feedback. Work by Salton (1972) has indicated that these facilities permit systems built on automatic indexing to out-perform systems such as MEDLARS based on manually constructed controlled indexes. As automatic keyword indexing is relatively easy and inexpensive to implement, this has obvious economic implications in the development of future information retrieval systems. Unfortunately Salton's work was based on a file organization which is not practical when applied to a large data base and indeed some doubt has been expressed as to the validity of Salton's results in view of the small size of the data base used in his experiments. Thus one purpose of the retrieval system described here is to incorporate the above features into a more conventional and efficient file organization in order to evaluate their usefulness using a sizeable data base.

A second aim of the Mistral system is to develop an interface-building system in which query and command languages can easily be implemented. The retrieval language has been built using an extendible

query analyser, (Leese and Macleod, 1974). This is, in effect, an extendible programming language. The addition of a new syntactic unit to the language is similar to function definition in a normal language except that there are also facilities for specifying the syntax of the new unit. Thus the basic programming language may be extended to incorporate features appropriate to some particular application. The semantics of a new language unit may be expressed in terms of the existing semantic capabilities, or these basic semantics may be themselves extended by the addition of 'primitive' functions, (written, in the current implementation, in Burrough's Algol). In this retrieval application a set of primitives have been developed which provide an interface between the existing language and a data management system, (Burroughs Corporation, 1974). This provides the basic retrieval system upon which the commands described below have been built. A Mistral command or other language unit is defined by a statement of the form:

commandtype = syntax specification

where "commandtype" will be the name of the command and "syntax specification" defines the structure of the command. The syntax is specified in a fairly straightforward meta-language the use of which should be clear from the examples below. The commands have a basic syntactic structure similar to that employed in many programming languages. Ideally, query languages, should be simple yet flexible. These two aims are, in a sense, somewhat contradictory. The most flexible type of language is one based on a programming language. Such a language is not however easily used by a person with little or no programming skills. In the language described below we attempt to reconcile these two goals by constructing commands which are composed of a large number of options. These options have programmable defaults. That is, the defaults can be preset in a user profile so that if the simplest forms of the command are used then the default actions taken can be chosen so as to meet the needs of the individual user. Once the user has had sufficient experience with the simple command forms to become aware of, and frustrated by, their limitations, he will in all probability become sufficiently motivated to learn some of the more complex options available in the command structure. In addition, Mistral is currently being extended to incorporate an extensive error recovery feature which will enable inexperienced users to make effective use of the system through a question-answer type of dialogue, (Macleod and Avis, 1975). This dialogue will also be designed so as to teach such users about the basic command structure. Thus the complete language is intended to satisfy the needs of casual users, inexperienced users who wish to expand their knowledge of the system, and also experienced users.

THE RETRIEVAL LANGUAGE

The language consists at present of three groups of commands. These are FIND, used to retrieve, LIST, used to produce output and USE,

which alters the default values of the options permitted in the other commands. This last command is in effect used to build a user profile. The complete syntax of the commands is given in the Appendix and their use is illustrated below.

The FIND command, which is the basic retrieval command, is defined as:

```
FINDCOMMAND = "FIND"(SEARCHOBJECT)/("LIKE"[INTEGER*","])
                ["IN" DATABASEXPR]
```

(Objects in quotes represent terminals of the language while unquoted names represent non-terminals whose syntax is defined elsewhere. Square brackets enclose options, parentheses enclose syntactic groupings and the slash symbol separates alternatives. The meta-symbol "*" indicates that the item preceding this symbol may be repeated indefinitely with each occurrence separated by the symbol following the asterisk. In this example the command contains either a SEARCHOBJECT or the LIKE phrase and the latter may optionally be followed by a list of integers separated by commas.)

A SEARCHOBJECT is used to describe the documents to be retrieved. It is made up from any combination of words using Boolean AND, OR and NOT connectives. A blank separating two words is an implicit OR. Parentheses can be used to indicate the precedence with which the operations should be applied. Otherwise precedence is left to right. Examples of search objects are:

```
"WHALES" AND "PLANKTON" AND "OIL"
"MIGRATION" AND ("MOOSE" OR "CARIBOU") BUT NOT "ALASKA"
```

The terms specified in SEARCHOBJECT are searched for in the inverted index. However the search may also be conditional on the contents of one or more fields in the document. This part of the search object is preceded by the word WITH followed by a list of field names and field contents. For example:

```
FIND "POLLUTION" WITH "ONTARIO" IN ABSTRACT AND DATE AFTER 1973
```

The LIKE alternative causes a search for documents similar to the ones whose document numbers are specified by a list of integers. (The LIST command provides the document numbers of any documents displayed.) If the list is omitted then a search is made for all documents similar to those indicated as being relevant since the occurrence of the last search. This provides the ability for a feedback search where the search terms are constructed from those contained in the relevant documents. Document relevance is indicated through the use of the LIST command.

The DATABASEXPR option allows the user to define the set of

documents to be searched. This may be the name of a data base or it may be some combinations of data bases or data bases subsets. If omitted the database used is the one established as the default through the USE command. A data base name may be a system data base name or the implicit data base called RESULT which contains the results of the last search, or the data base UPDATE which contains the document numbers of the most recent update to the collection, or a data base name created by the user. For example:

```
FIND "MINE" AND "ACID" AND "DRAINAGE" WITH "CANADIAN"
IN ABSTRACT, DATE > 1970 IN POLLUTION
```

This will search the data base "POLLUTION" for all documents indexed under "MINE", "ACID" and "DRAINAGE". Only documents published after 1970 and with the word "CANADIAN" in the abstract will be retrieved.

The FIND command outputs the number of documents retrieved. If the command contains only index terms, then the exact number of documents to be retrieved is known from the index files. In this case the actual number of relevant documents found is output. For example: "50 DOCUMENTS FOUND". If the command also contains field searches, such as "WITH "CANADIAN" IN ABSTRACT" then the actual documents must be retrieved and examined. In this case the output will be the number of documents to be searched. For example: "50 DOCUMENTS TO BE SEARCHED". Since the actual retrieval and examination of documents will be more expensive for a large number of documents, the user has the possibility of modifying his search again before the actual retrieval is performed.

The LIST command generates output which may be a listing of documents, or part of the index or thesaurus, or a report on the status of the user's profile. When documents are listed these are from the RESULT data base unless some other data base is explicitly specified. Output may be on the current display device or the printer, or a named data base may be created. The user may also specify which documents and fields are to be listed as well as the ordering of the documents. The normal default ranking is on relevance weights.

Each document displayed on the user terminal is followed by the system query "RELEVANT?". The next document is obtained by typing one of "Y" or "N" or by pressing the return key. This feature is used to gather statistics on the validity of the current document indexes and may later be used in automatic index modification. It also provides feedback for the "FIND LIKE" command. Each document display includes the document number which is the positional index of that document within the system data base. Examples are:

```
LIST
```

which causes documents to be listed according to the default options

indicated in the user profile:

```
LIST 100 TITLES AND ABSTRACTS RANKED BY TITLE ON PRINTER
```

lists the titles and abstracts fields of the first ten documents on the printer. The output is alphabetically ordered on titles:

```
LIST 1 TO 10,  45 TO 50 TITLES ON MYBASE
```

which takes the first ten titles, skips the next thirty-five and then takes the next six, and places them in a new data base called MYBASE.

The INDEX option allows the user to browse through the data base using the index. For example:

```
LIST INDEX "CYANIDE" WITH THESAURUS
```

causes the index term "CYANIDE", (or the alphabetically closest index term if this particular term does not exist), to be listed together with the list of all the document numbers of the items it references. The inclusion of the THESAURUS option causes the corresponding entries in the thesaurus for this particular term together with the associated document numbers to be listed. This feature together with the FIND LIKE command provides a fairly powerful browsing facility.

The LIST STATUS command provides a status report on the current user profile while the USE command records in the profile, defaults for options which may occur in the other commands. For example, it may specify the data base used in searching or listing, the fields of documents to be displayed, the device to be used for listings and the ranking algorithm to be used for output. It may also be used to give names to groups of search terms or entire commands so that the user may save particular commands for re-use at later searches without extensive retyping. Finally it governs control of the thesaurus.

SUMMARY

The language described above is intended to illustrate how a retrieval language with a range of capabilities can be designed in a reasonably systematic manner using an extendible command analyser. As experience in developing retrieval systems increases it is becoming ever more apparent that overly simplistic retrieval languages are frequently inappropriate search tools. Features such as extensive boolean search capabilities, flexibility in listing and formatting of output, feedback searches and control of thesaurus usage are among those which seem desirable in a retrieval system. Appropriate error recovery features can also be incorporated to make the system accessible to the unskilled user. This list is by no means exhaustive, yet even these few features require a fairly complex command language if they are to be used

effectively. An ad hoc approach to the user interface design cannot result in a flexible concise language. It is this author's belief that the use of a command analyser such as the Mistral system greatly simplifies language design and implementation.

APPENDIX - THE LANGUAGE SYNTAX

```
FINDCOMMAND   = "FIND" ("LIKE" [INTEGER * ","])/SEARCHOBJECT
                ["IN" DATABASEXPR]
SEARCHOBJECT  = SEARCHEXPR ["WITH" SEARCHFIELDS] / NAME
SEARCHEXPR    = (SEARCHEXPR ("AND" / (["BUT"] "NOT")/
                ["OR" SEARCHTERM)...)/SEARCHTERM
SEARCHTERM    = STRING / ("("SEARCHEXPR")")
SEARCHFIELDS  = SEARCHFIELD * ("AND"/",")
SEARCHFIELD is data base dependant.  Possible forms are:
SEARCHFIELD   = (["DATA"("AFTER"/ ">")/"BEFORE"/ "<") STRING)/
                ("AUTHOR"STRING)/("TITLE"/"ABSTRACT" "CONTAINING"STRING)
DATABASEXPR   = (DATABASEXPR "IN"/(["BUT"]"NOT")/["WITH"] DATABASETERM)/
                DATABASETERM
DATABASETERM  = (DATABASENAME [REFNOS])/("("DATABASEXPR")")
REFNOS        = (INTEGER "TO" INTEGER)/(INTEGER)
LISTCOMMAND   = "LIST" DOCLIST / INDEX / "STATUS"
DOCLIST       = [REFEXPR * ","][FIELDS] ["OF" DATABASEXPR]
                ["ON" DEVICENAME] ["RANKED"["BY"] RANKALG]
REFEXPR       = REFNOS / SPEC
SPEC          = ("NEXT" [INTEGER])/("PREVIOUS" [INTEGER])
FIELDS        = FIELDNAME * "AND"
INDEX         = "INDEX" STRING [SPEC]["WITH" "THESAURUS"]
USECOMMAND    = "USE" DATABASE/FORMAT/DEVICE/RANK/LIKE/THESAURUS
DATABASE      = DATABASEXPR ["IN" "LIST" / "FIND" ["COMMAND"]]
FORMAT        = [REFEXPR * ","] FIELDS/"ALL" "AS" "FORMAT"
DEVICE        = DEVICENAME "TO" "OUTPUT"
RANK          = "RANKING" "BY" RANKALG
RANKALG       = "WEIGHT"/(["ASCENDING"/"DESCENDING"] FIELDNAME)...
LIKE          = NAME "FOR" (DATABASEXPR/SEARCHOBJECT/""COMMAND"""")
THESAURUS     = ["NO" / "NARROW" / "MEDIUM" / "BROAD"] "THESAURUS"
```

REFERENCES

Burroughs Corporation, 1974, DMS 2 Data Management System, Burroughs Corporation.

Leese, K.R. and Macleod, I.A., 1974, An Extendible Command Analyser, INFOR Journal, 12(2), 142-162.

Macleod, I.A. and Avis, J.C., 1975, Error Recovery for Casual Users of Query Languages, Technical Report (in preparation), Department of Computing Science, Queen's University.

Salton, G., 1972, A New Comparison Between Conventional Indexing and Automatic Text Processing, J.A.S.I.S., 23(2), 75-84.

NEW SYSTEMS AND THEIR HUMAN DIMENSIONS IN AN ACADEMIC LIBRARY (SYSTEMES NOUVEAUX ET LEURS DIMENSIONS HUMAINES DANS UNE BIBLIOTHEQUE UNIVERSITAIRE)

W.R.M. Converse and O.R. Standera
University of Calgary
Calgary, Alberta T2N 1N4

ABSTRACT

This paper briefly describes systems currently operational at the University of Calgary. It then examines the impact of these systems, services and technologies on both users and librarians. The paper is based on the experience of the University of Calgary Library with automated and semi-automated systems which also provides the basis for the generalizations which the authors make. (Ce mémoire est une courte description des systèmes actuellement en marche à l'Université de Calgary. Il examine l'effet de ces systèmes et ces technologies sur le public aussi bien que sur les bibliothécaires. L'étude repose sur l'expérience qu'a eue la bibliothèque de l'Université de Calgary de systèmes automatisés en tout ou en partie, expérience qui sert de base aussi aux généralisations formées par les auteurs.)

INTRODUCTION

Since 1971 The University of Calgary Library has introduced a number of automated and semi-automated systems. These systems are operational in the areas of administration, acquisitions, cataloguing, collections development, periodicals, circulation and information retrieval. In addition, the Library now has the capability of accessing some twelve million bibliographic records on-line from outside suppliers. The implementation of all these systems has had a considerable impact on the Library organization as a whole, evidenced by the new and different information activities undertaken and contemplated, new user services, data bases and participation in national and regional projects. From among the several areas affected by these changes, including costs, internal structures, and external relationships, the authors have singled out the human element, namely, users and librarians, to focus on, applying their experience gained at The University of Calgary as well as their observation of trends traceable elsewhere.

SYSTEMS CURRENTLY OPERATIONAL AT THE UNIVERSITY OF CALGARY LIBRARY

There are six systems currently operational at The University of Calgary. These are the EXCON System, an accounting system; the TESA