Information Technology: Research and Development (1982), *2* (247–260)

A SIMPLE, INTELLIGENT FRONT END FOR INFORMATION RETRIEVAL SYSTEMS USING BOOLEAN LOGIC

M. H. HEINE

School of Librarianship & Information Studies, Newcastle upon Tyne Polytechnic, Newcastle upon Tyne NE1 8ST, UK

(*Received 29 March 1982, revised 2 July 1982*)

ABSTRACT

It is argued that it is unnecessary to require the user of an information retrieval system to characterize his information need in both semantic and syntactic (Boolean) manners, as is the case with conventional systems. A simple sub-system acting as a front end to conventional systems can eliminate the need for user-specified Boolean characterization, and in addition allow for relevance-feedback to improve the semantic characterization. The method suggested involves (1) the construction and ranking (invisibly to the user) of a set of elementary logical conjuncts of record attributes, and (2) their stepwise, dynamic inputting to the conventional system, and the reordering of the elementary logical conjuncts, based on user-specified relevance feedback, using a novel method: linear discriminant analysis based on simple Bernoulli variables, with prior reselection of record attributes. The front end so described could in principle be part of the same computer program that drives the retrieval system, or it could be independent of the latter and implemented in an intelligent terminal interacting with a remote system.

1. INTRODUCTION

The essential and in part defining feature of information retrieval is its fallibility. If retrieval from a database involves foreknowledge of a set of record attributes that are attached to relevant records and to no other records, then the procedure is (conceptually) a trivial one. The problem becomes non-trivial, indeed has become an area of considerable academic interest, for two reasons:

1. Such a set of record attributes may not exist (i.e., for every set of record attributes, there may be non-relevant records as well as relevant records characterized by the set).
2. Foreknowledge of the type described does not exist (i.e., if there is such a set of attributes, it is an unknown).

The picture we currently have of information retrieval is as follows. An *information need* (recognized subjectively by an individual person, and therefore not knowable scientifically, i.e., a primitive entity) is the prompt for an *information search.* In the latter, objective or observable process, an enquirer in effect maps his information need to a set of *record attributes.* The mapping process is fallible, whether the enquirer recognizes it or not. This fallible characterization of the information need is assumed to be such that, by some matching process, the records that are *relevant* to it (i.e., which will communicate information so as to meet the need) will be obtained, and those that are not relevant to it will be rejected. The 'target set' of records is only perceived, not known, and this set of attributes is a formal description of that perception. The most important features of the retrieval process, so described, are

1. The enquirer's perception is a fallible one—he can only *guess* at the record attributes that will be most effective.
2. The 'matching' process referred to is usually one based on the construction of a logical expression (the 'search statement') which determines retrieval of a record when, and only when, it assumes the value 'TRUE' for that record.

The enquirer is thus doubly burdened: he must guess at the optimal set of record attributes (which entails deciding on both the *size* of such a set, as well as the *identity* of the attributes), and he must guess at the form of the logical expression which will, most effectively for him, retrieve records in an optimal manner. These two characterizations of his information need might sensibly be referred to as semantic and syntactic characterizations, respectively. One might reasonably claim that the burden is unreasonably heavy: a user may be confused just by having two problems; he may also be confused by his insufficiently distinguishing between his own verbal notions of what his 'question' is (in terms of his own private vocabulary) and what the question is as an input (*qua* search statement) to a retrieval system. He may additionally be performing well in one role but not the other without realizing it, for example identifying useful attributes well, but constructing ineffective logical search expressions.

The notes that follow attempt to offer, first, a procedure allowing the user to concentrate on semantic characterization only (the syntactic role being undertaken in an invisible and optimum manner *by the system*), and secondly, a procedure for redefining the attributes using relevance feedback (i.e., redefining the semantic characterization), this time invisibly. These two procedures are described in common in that they are end-on (and interactive) and in that both of them involve the construction, ranking and stepwise inputting to a conventional retrieval system, of the members of a set of elementary logical conjuncts of record attributes. The procedures are simple both to program and implement, and should impose little cost overheads—whether incorporated in a conventional retrieval system or in an intelligent terminal interfacing a conventional system but remote from it.

2. THE CONVENTIONAL RETRIEVAL PROCESS AND ITS WEAKNESSES

From the users' point of view, most retrieval systems operate as follows:

1. An enquirer postulates a set of record attributes, for example, the terms ARTERY, CEROID and RUSSIAN, which in effect offer a characterization of

a recognized information need. (An alternative, equivalent viewpoint is that an information *channel* is thereby set up, between the enquirer and the creators of the individual records. The channel is only partly characterized at this stage.) If the record attributes are terms, then the characterization could be said to be a semantic one.

2. Each of the record attributes is then identified with an elementary logical variable (ELV) which, perhaps confusingly, is usually given the same name as the attribute. The ELVs here are Boolean variables taking just one of the two values 'TRUE' or 'FALSE' for each record in the database. An ELV evaluates to TRUE if and only if the record concerned contains the equivalent attribute. If the attribute is a date of publication, D, on the other hand, and the enquirer has specified that this must be later (say) than some value T, then the ELV concerned will be an inequality such as $D > T$. (A full and systematic analysis of the use of cardinal-valued variables in information retrieval has been offered by Lipski and Marek, 1979.) Thus the attribute 'ARTERY' is identified with an ELV 'ARTERY', etc. To distinguish attribute from related ELV we will henceforth denote ELVs by boldface type, e.g., **ARTERY**.
3. The enquirer then constructs a logical search statement (or 'profile' or 'query', though the latter term is used so ambiguously it is better avoided). This is of the form of a logical expression, the ELVs of which are just the ELVs described above. For example, the search statement might be:

(**CEROID** OR **ARTERY**) AND NOT **RUSSIAN**
or (**CEROID** AND NOT **RUSSIAN**) OR **ARTERY**.

The search statement represents a syntactic characterization of the information need concerned (or alternatively it gives a syntactic characterization to the information channel that has—incompletely—been set up previously).

4. The logical search statement so formed is then input to a system that implements a record-retrieval algorithm. Invisibly to the user, the system evaluates the search statement for each record in the database. (If, for example, a record structure includes the terms CEROID, but not the terms ARTERY or RUSSIAN, then the ELVs **CEROID**, **ARTERY** and **RUSSIAN** evaluate to TRUE, FALSE and FALSE, respectively, and the search statements given above evaluate to TRUE and TRUE, respectively.) The system then outputs (or at least makes available for output) the records that evaluate the search statement to TRUE. There may be some fine-tuning here: the record set may be sorted in some way first, and/or the size of the output may be limited in some *ad hoc* way, but the basic procedure is as described.

The above description is the usual basic one, irrespective of whether

1. The database itself is a sequential structure of records, or a sequential or tree structure of attributes with addresses of records attached (i.e., an inverted file structure).
2. Retrieval is conducted on- or off-line.
3. The system output is despatched to a disk file or line printer.

The most critical weaknesses of the basic procedure are, it is suggested:

1. The enquirer is burdened with *both* semantic characterization *and* syntactic characterization of the information channel that he creates.
2. There is no systematic provision of a procedure to use feedback to the system to improve (automatically) the effectiveness of the channel: a procedure that could usefully be labelled 'auto-interactive'.

Investigation of this second possibility has been carried out experimentally and theoretically since the mid-1960s (e.g., Rocchio and Salton, 1965; Rocchio, 1966) as reviewed by Salton (1975), with conceptual roots perhaps back to Kochen (1962). More recently the problem has been treated by Yu *et al.* (1976), Vernimb (1977), Dosckocs (e.g., Dosckocs and Rapp, 1979), Harper (1980) and Attar and Fraenkel (1981). Dosckocs describes the development of a promising operational auto-interactive system, but this is still at the prototype stage and the description he offers of it is only indicative. The closest to an operational auto-interactive system is perhaps Porter's 'CUPID' system, now running experimentally at Cambridge and Dublin (Porter, 1982). The theoretical background to Porter's work is described in van Rijsbergen *et al.* (1980), following on from van Rijsbergen (1977). The system devised by Attar and Fraenkel appears to be at a comparable stage but relies for its feedback action on clustering of all retrieved documents, rather than on a contrasting (e.g., of clusters) of retrieved-relevant and retrieved-not-relevant documents.

There is no major operational system of the standing of say DIALOG or SDC which offers auto-interactive retrieval to its users, i.e., which allows user-specified data on the relevance of retrieved records to improve retrieval performance. However it may reasonably be expected, given the pace of theoretical developments and computing capability, that this type of service will be routinely available to users by the end of the decade.

The following notes attempt to clarify and remedy the two weaknesses in conventional retrieval we have identified. The feedback algorithm is believed to be a novel one in this area of application. In effect the procedures described constitute an intelligent front end to conventional retrieval systems.

3. AUTOMATIC SYNTAX SYNTHESIS

The revised retrieval procedure suggested is such that *the system user specifies not a logical search statement but a set of attributes.* (In this respect the system described is similar to Porter's CUPID system.) In its simplest form, the system-user dialogue would be (for example):

system:	INPUT SEARCH TERMS	} D1
user:	CEROID, ARTERY, PIGMENT	}

where it is understood that

1. *No* logical variables or operators are explicitly given by the user.
2. The comma (or some substitute character) serves simply to separate attributes—it is not a symbol for a logical operator.

No attempt is made by the user to create a logical search expression. The fallibility of the input is thus reduced in the sense that commitment to a particular search expression has yet to be made, and on the assumption that the means to identify an optimum expression will be forthcoming.

In a variant of the above dialogue, the system could prompt the user not for a *set* of attributes but for a *sentence* characterizing the information need (channel). A simple, pre-system routine could then strip the sentence of articles, prepositions and other stop words, perhaps truncate the remaining words (to reduce word variability due solely to inflexion), and thereby arrive at a set of attributes similar to that given in dialogue D1 above.

A further dialogue is then necessary, in which the system prompts the enquirer for an upper limit to either the number of records to be retrieved, or the fraction of all documents that are relevant to the information need and which are to be retrieved, i.e., an upper limit to (estimated) recall. For example, we might have the dialogues:

system: INPUT MAXIMUM NO. OF RECORDS TO BE RETRIEVED
user: 80 } D2

or

system: INPUT MAXIMUM REQUIRED RECALL LEVEL
user: 0.6 } D3

respectively. The system will be able to apply the first criterion infallibly, but the second only fallibly since appeal to a model of retrieval will need to be made in estimating the recall level for a given search and response.

The system will then execute the following steps, all of which can be invisible to the user. We shall adopt the notation FES for 'front end (to) system' and RS for the residual system. (A conventional retrieval system as described in Section 2 thus becomes an RS system.) Lastly, we shall refer to an FES that inputs a maximum value to the number of records, i.e., as in dialogue D2, as an FES.A system, and one that inputs a maximum value for recall, i.e., as in dialogue D3, as an FES.B system. The notation $T1, T2, \ldots$ for attributes, and $\boldsymbol{T1}, \boldsymbol{T2}, \ldots$ for the corresponding ELVs will also be used. The notation $\overline{\boldsymbol{Ti}}$ denotes an ELV equivalent to the logical expression NOT $\boldsymbol{Ti}$.

1. Construct all elementary logical conjuncts (ELCs) of the input ELVs. For n input attributes: $T1, T2, \ldots, Tn$, there will be 2^n ELCs.

   For example, if three record attributes are specified by the user: $T1$, $T2$ and $T3$, the 8 ELCs that will be so constructed are:

$$\boldsymbol{T1} \wedge \boldsymbol{T2} \wedge \boldsymbol{T3}$$
$$\boldsymbol{T1} \wedge \boldsymbol{T2} \wedge \overline{\boldsymbol{T3}}$$
$$\boldsymbol{T1} \wedge \overline{\boldsymbol{T2}} \wedge \boldsymbol{T3}$$
$$\boldsymbol{T1} \wedge \overline{\boldsymbol{T2}} \wedge \overline{\boldsymbol{T3}}$$
$$\overline{\boldsymbol{T1}} \wedge \boldsymbol{T2} \wedge \boldsymbol{T3}$$
$$\overline{\boldsymbol{T1}} \wedge \boldsymbol{T2} \wedge \overline{\boldsymbol{T3}}$$
$$\overline{\boldsymbol{T1}} \wedge \overline{\boldsymbol{T2}} \wedge \boldsymbol{T3}$$
$$\overline{\boldsymbol{T1}} \wedge \overline{\boldsymbol{T2}} \wedge \overline{\boldsymbol{T3}}$$

   where, at this stage, the ELCs are in an arbitrary order.

2. The ELCs so formed then have a real numerical value attached to them, using one or other sub-procedure, and with the choice of sub-procedure influenced by some pre-specified univariate measure of retrieval effectiveness. At its simplest, we could attach the value of the following expression:

$$\sum_{i=1}^{n} f(Ti)$$

to each ELC, where $f(Ti)$ takes the value 1 when ***Ti*** denotes a non-negated ELV and the value 0 otherwise.*

The ordering so achieved can be strong or weak. When it is weak (i.e., several of the ELCs bear the same numeric value) then order the ELCs that bear common values locally arbitrarily.

Label the ELCs so ordered by ***L1, L2,*** . . . ***LJ***, where $J=2^n$, ***L1*** having the highest numeric value attached to it, ***LJ*** the lowest. *All* mappings of the ELCs to the real numbers should give the *all-negated* ELC (for a given set of attributes) the *lowest* numerical value, i.e., make this ELC the one labelled ***LJ***.

3. If the system is of type FES.A then proceed as in 3.1; if it is of type FES.B proceed as in 3.2:

3.1 Input ***L1*** to the RS. Count the number of records retrieved, i.e., which evaluate ***L1*** to TRUE. If this number does not invalidate the user-specified criterion, then input ***L2*** and concatenate the records so retrieved with those already retrieved. (There will be no overlap in these retrieved sets, since the ELCs partition the database.) Apply the criterion again and iterate, but always exclude ***LJ*** as input from the RS since this ELC will retrieve most of the database.† (Concatenation of the records retrieved for ***LJ*** with the records retrieved for the previously-input ELCs will retrieve *all* of the database.)

3.2 Proceed as in 3.1 above but use a *modelling* distribution over the ELCs to evaluate (fallibly) the criterion concerned.

Some further comments, some speculative, on the sequence of steps described above are now offered.

1. When an ELC is to be input to an RS, it may be advantageous to *order* the component logical variables first, with non-negated ELVs preceding negated ELVs, and with non-negated ELVs associated with more specific (i.e., 'rarer')

* This expression is identifiable with the Cranfield measure 'level of coordination'. A record in the database will evaluate exactly one of the ELCs to TRUE. The ELC concerned will have each of its component (negated or non-negated) ELVs evaluating to TRUE. The number of non-negated ELVs in this case is the record's coordination level.

† Extensive experimental work by Heine (1981) indicates that for information needs in medicine, for the database MEDLARS, and for $n=5$, the probability that a non-relevant [relevant] document record maps to the all-negated ELC is 0.987 [0.252]. Obviously the probabilities concerned will vary with the nature of enquirers' information needs, the database concerned, the skills of enquirers at characterizing their needs (i.e., how good they are at identifying 'useful terms'), and the value of n.

terms preceding ELVs of more common terms. This will prevent unnecessary overload of the RS, since the shortest lists of records will be formed first. (This is assuming that the RS acts on ELVs from left to right.)

2. The useful upper limit on n may be about five (i.e., 32 ELCs). This suggests that the user should be encouraged to study carefully any hierarchies of record attributes before inputting a set of attributes to the system; 'coarser' generic terms may then usefully substitute for the more specific terms that they disjoin.
3. Despite the assumption we have broadly made, that the user should concentrate his efforts on identifying useful attributes, and ignore syntactic, i.e., logical, characterization, it may be useful to recognize a need for obligatory logical variables in the search expressions to be input to the RS. The user may for example be utterly convinced (rightly or wrongly) that the ELV **RUSSIAN** should be given such status in respect of some need, while agreeing that all other attributes he has specified are more 'permissive' or flexible in their syntactic status. (We use here the vocabulary 'obligation' and 'permission' of the 'Deontic' logicians, e.g., al-Hibri, 1978.) In this case, after appropriate system prompting, the ELCs would be reduced in number before steps 2 and 3 (p. 252) were undertaken. The reduction would take place as follows. Suppose the user specified the attribute $T3$ as being such that it must *always* [never] appear in records to be retrieved. Then the FES would *reject* all ELCs containing $\bar{T}3[T3]$ respectively, before going to step 2. The use of deontic ELVs would however be exceptional—by default users would not be prompted by the system to use them since this would simply entail taking the system back to a non-intelligent, conventional system in which the user bore the syntactic burden and the penalties that go with it.
4. We repeat that where a user specifies a quantitative attribute (e.g., the date of preparation of a record), the appropriate ELV will be of the form of an inequality, e.g., $T1 := (\text{AGE} < 23)$, where each record has a field, named AGE, containing a numeric value that enables $T1$ to evaluate to TRUE or FALSE. The prompt and response dialogue:

system:	QUANTITATIVE ATTRIBUTES?	D4
user:	AGE < 23	

might be appropriate to this case.

4. HEURISTIC PROCEDURES

In this section we discuss the role of an FES in contesting, and improving the user's specification of record attributes. The algorithm given is novel (in this area of application of the technique concerned) and has not yet been tested experimentally. However, the technique is well established in applied statistical work (see, e.g., Gilbert, 1968) and it is accordingly suggested that it is *prima facie* valid in information retrieval. Unlike some earlier published techniques, the one to be described is simple, easily programmable, and should be quickly computable.

The approach represents an application of the discrete Swetsian formalism as extended by Heine (1981). It explicitly recognizes term-dependence. Indeed the existence of pairwise dependence is the basis of the heuristic procedure, such dependencies being examined in both the set of relevant documents retrieved and the

set of non-relevant documents retrieved.* Our approach will be essentially that of applying linear discriminant analysis (see Hoel, 1971, for example) to Bernoulli random variables defined for each term in partitionings of a retrieved set, the dependencies just defined appearing as covariances between these variables. First, we denote the set of retrieved documents by W_3, the set of relevant documents retrieved by W_1, and the set of non-relevant documents retrieved by W_2, so that $W_1 \cup W_2 = W_3$ and $W_1 \cap W_2 = \emptyset$. Denote a document by d, the set of attributes making up the query by $\{Tp\}$ (so that $n = \| \{Tp\} \|$), and the three random variables associated with an attribute Tp and sets W_i $(i = 1, 2, 3)$, by X_{pi}. Each of the random variables X_{pi} is a Bernoulli variable, since it maps each member of the set concerned to one of two values, commonly chosen to be 0 and 1. There are $3n$ random variables so defined. Thus:

$$X_{pi} = \begin{cases} 1 & \text{if } d \varepsilon W_i \text{ and } d \text{ is assigned attribute } Tp \\ 0 & \text{if } d \varepsilon W_i \text{ and } d \text{ is not assigned attribute } Tp. \end{cases}$$

The probability of the event $X_{pi} = 1$ is accordingly just the fraction of the documents in W_i that are assigned attribute Tp. (X_{p3} is thus an estimate, based on the sample W_3 of the entire database, of the specificity of Tp but the estimate may be a biased one.)

The problem we now set ourselves is this: What linear function of the random variables X_{p3} (n in number), itself a random variable, will yield a successor retrieved set more effective than the predecessor retrieved set W_3? To solve this problem define:

$$Z_i = \sum_{p=1}^{n} \lambda_p X_{pi} \qquad (i = 1, 2, 3)$$

the third of these random variables, Z_3, being the function intended to meet the objectives just stated. In essence, the problem is to obtain coefficients λ_p that are 'most effective'. (We note in passing that when $\lambda_p = 1$ (all p) the Z_i measure simple level-of-coordination (for all retrieved documents if $i = 3$, for relevant-retrieved documents if $i = 1$, for non-relevant-retrieved documents if $i = 2$). Also that when, additionally, the variables X_{pi} are independent and identically distributed for all p, the variable Z_i is binomial.) In order to apply Fisher's discriminant analysis technique to this problem, we also define variables h_{pi}, S_{pq} and D_p as follows:

1. h_{pi} is the parameter of X_{pi} (so that $E(X_{pi}) = h_{pi}$; $V(X_{pi}) = h_{pi}(1 - h_{pi})$)

   where $E(\ldots)$ and $V(\ldots)$ denote expectation and variance.
2. For two attributes T_p and T_q:

* By 'documents retrieved' we are of course briefly signifying 'records retrieved denoting documents that are relevant.'

$$S_{pq} = \sum_{i=1}^{2} || W_i || E[(X_{pi} - E(X_{pi}))(X_{qi} - E(X_{qi}))] \text{ (definition)}$$

$$= \sum_{i=1}^{2} || W_i || \text{Cov}_i (X_{pi}, X_{qi})$$

That is, S_{pq} is defined as a weighted sum of the covariances of random variables defined for the two attributes, and for the sets W_1 and W_2, the weights reflecting the relative sizes of the two subsets of W_3. (The subscript to Cov denotes the set of documents, either W_1 or W_2, that is the common domain of the two random variables concerned.) In particular we note:

$$S_{pp} = \sum_{i=1}^{2} || W_i || V(X_{pi})$$

$$= \sum_{i=1}^{2} || W_i || h_{pi}(1 - h_{pi})$$

3. The signed difference in the means of the two distributions induced by X_{p1} and X_{p2} is denoted by D_p, i.e.,

$$D_p = E(X_{p1}) - E(X_{p2})$$

and the vector of such values for all n attributes is denoted $\underline{D}$:

$$\underline{D} = (D_1, D_2 \ldots\ldots, D_n)$$

The linear discriminant analysis algorithm was identified by Fisher in entailment of the maximization of the expression: $(E(Z_1) - E(Z_2))^2/(V(Z_1) + V(Z_2))$. This is one estimate of the separation of the populations of which W_1 and W_2 are samples. In a retrieval context, the expression can be identified with the square of a measure of retrieval effectiveness suggested by Brookes (1968). The algorithm involves defining the $n \times n$ matrix $(\lambda_p S_{pq})$, and solving the n component equations of $(\lambda_p S_{pq}) = \underline{D}^T$ for the values of λ_p. These values, when inserted into the expression for Z_3, allow the ELCs to be optimally ordered by the weight:

$$\sum_{p=1}^{n} \lambda_p f(\boldsymbol{Tp})$$

where $f(\boldsymbol{Tp})$ is as earlier defined (i.e., $f(\boldsymbol{Tp}) = 1[0]$ if Tp is a non-negated [negated]

ELV). So ordered, the ELCs may then be input to the RS until some stopping criterion is met.

A full treatment would include cases where $E(X_{ij})$ and $V(X_{ij})$ can be indeterminate (where the vector of occurrences can be a string of 0s), which will occur for very good or very bad terms. (The latter should however not appear in $\{\boldsymbol{Tp}\}$.) Also, as pointed out by an anonymous referee, there may be terms $\boldsymbol{Tp}$ for which $S_{pq} = 0 = S_{pp}$ for all q. The author's view is that this makes the equations in λ_i underdetermined rather than incorrect, and that *ad hoc* branches within the algorithm should be taken when such exceptional conditions are met.

What we did not specify in the above description is the origin of the attributes $\boldsymbol{Tp}$. These may be the same attributes as those specified by the user in arriving at the predecessor retrieved set, W_3. That set may have been the *first* retrieved set arrived at by the steps described in Section 3, or it may have been determined by a previous application of the technique just described. But the attributes could have been arrived at otherwise: they could, for example, have been obtained by choosing the most frequent attributes in W_3, or by ranking attributes by the value of $E(X_{i1})/E(X_{i2})$ and choosing the top-ranking attributes, or by a more sophisticated approach taking attribute dependencies into account, e.g., by clustering the attributes in W_3 and selecting those that are shallowest in the clustering (as investigated in another context by the author). However, it is unprofitable to speculate further on the best decision here: only an experimental approach can be conclusive.

4.1 Example

To clarify the use of the technique described above, we consider an example of a (predecessor) retrieved set which a user has partitioned into relevant and non-relevant records. The set is of 14 records, five of which are relevant. Four attributes have been identified as optimum for finding an improved (successor) retrieved set, let us say by ranking terms in the retrieved set by $E(X_{i1})/E(X_{i2})$ and taking the top four. (A procedure for stripping terms off records, and undertaking this computation is thus also needed, unless the terms originally specified by the user (Section 3) are retained.) The records are labelled A to N as shown in Table 1, so that $W_1 = \{A,B,C,D,E\}$, and $W_2 = \{F,G,H,I,J,K,L,M,N\}$.

There are thus eight Bernoulli variables involved, four for each partition of the retrieved set. For each random variable we can calculate an expectation for the set

Table 1. Trial data for a predecessor set of retrieved documents partitioned by relevance-judgements, and notation

Attribute	*Bernoulli variable*	*Relevant documents retrieved*					*Bernoulli variable*	*Non-relevant documents retrieved*								
		A	*B*	*C*	*D*	*E*		*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*
T1	X_{11}	0	1	0	1	0	X_{12}	1	0	0	0	1	0	1	0	0
T2	X_{21}	1	0	1	1	0	X_{22}	0	1	1	0	0	0	0	1	0
T3	X_{31}	0	1	1	0	1	X_{32}	0	0	0	0	0	0	1	0	1
T4	X_{41}	0	0	1	0	1	X_{42}	0	0	0	0	0	1	0	0	0
		$\|\| W_1 \|\| = 5$						$\|\| W_2 \|\| = 9$								

concerned, and for each pair of random variables in *one* of the two sets we can calculate a covariance. For example,

$$E(X_{21}) = 3/5, \quad E(X_{41}) = 2/5$$
$$\mathrm{Cov}_1(X_{21}, X_{41}) = E\,[(X_{21} - E(X_{21}))(X_{41} - E(X_{41}))]$$
$$= E\,[(X_{21} - 3/5)(X_{41} - 2/5)]$$
$$= 1/5\,[(0-3/5)(0-2/5) + (0-3/5)(1-2/5) + 2(1-3/5)(0-2/5) + (1-3/5)(1-2/5)]$$
$$= -1/25$$

The calculation of covariances is made clearer if we represent frequencies of co-occurrences of events in a 2×2 table. For X_{21} and X_{41}, for example, the table is:

X_{41} \ X_{21}	0	1	
0	1	2	3
1	1	1	2
	2	3	5

Similarly we can evaluate $\mathrm{Cov}_2(X_{22}, X_{42})$ to $-1/27$, and accordingly:

$$S_{24} = S_{42} = 5(-1/25) + 9(-1/27) = -8/15$$

The weighted sums of variances are similarly obtained. For example, since:

$$V(X_{21}) = 3/5(1-3/5) = 6/25, \quad V(X_{22}) = 3/9(1-3/9) = 2/9$$

we have:

$$S_{22} = 5(6/25) + 9(2/9) = 16/5$$

Also, the D_p values are readily found:

$$\text{e.g., } D_2 = E(X_{21}) - E(X_{22}) = 3/5 - 3/9 = 4/15$$

In full, the constants required are as follows:

$S_{23} = S_{32} = -22/15$, $S_{14} = S_{41} = -17/15$, $S_{24} = S_{42} = -8/15$, $S_{34} = S_{43} = 26/45$,
$S_{12} = S_{21} = -6/5$, $S_{13} = S_{31} = 2/15$, $S_{11} = 16/5$, $S_{22} = 16/5$, $S_{33} = 124/45$,
$S_{44} = 94/45$. Also, $D_1 = 1/15$, $D_2 = 4/15$, $D_3 = 17/45$, $D_4 = 13/45$

These are to be included in the equations:

$$\begin{aligned}
\lambda_1 S_{11} + \lambda_2 S_{12} + \lambda_3 S_{13} + \lambda_4 S_{14} &= D_1 \\
\lambda_1 S_{21} + \lambda_2 S_{22} + \lambda_3 S_{23} + \lambda_4 S_{24} &= D_2 \\
\lambda_1 S_{31} + \lambda_2 S_{32} + \lambda_3 S_{33} + \lambda_4 S_{34} &= D_3 \\
\lambda_1 S_{41} + \lambda_2 S_{42} + \lambda_3 S_{43} + \lambda_4 S_{44} &= D_4
\end{aligned}$$

yielding:

$$\begin{aligned}
48\lambda_1 - 18\lambda_2 + 2\lambda_3 - 17\lambda_4 &= 1 \\
-18\lambda_1 + 48\lambda_2 - 22\lambda_3 - 8\lambda_4 &= 4 \\
6\lambda_1 - 66\lambda_2 + 124\lambda_3 + 26\lambda_4 &= 17 \\
-51\lambda_1 - 24\lambda_2 + 26\lambda_3 + 94\lambda_4 &= 13
\end{aligned}$$

Solving for the four unknowns gives: $\lambda_1 = 0.234$, $\lambda_2 = 0.329$, $\lambda_3 = 0.242$, $\lambda_4 = 0.283$. Assuming these values to be the best estimates of the comparable coefficients that would serve to discriminate the set of *all* relevant documents from the set of *all* non-relevant documents, the optimum weighting function is thus found to be:

$$Z_3 = 0.234\, X_{13} + 0.329\, X_{23} + 0.242\, X_{33} + 0.283\, X_{43}$$

To obtain a successor retrieved set based on the attributes $T1$, $T2$, $T3$ and $T4$, we then rank the ELCs associated with $T1$, $T2$, $T3$ and $T4$ using the weighting expression:

$$Z = 0.234\, f(\boldsymbol{T1}) + 0.329\, f(\boldsymbol{T2}) + 0.242\, f(\boldsymbol{T3}) + 0.283\, f(\boldsymbol{T4})$$

where $f(\boldsymbol{Ti})$ evaluates to 1 when $\boldsymbol{Ti}$ evaluates to TRUE, and to 0 otherwise. The results of this ranking are given in Table 2.

Table 2. Elementary logical conjuncts (ELCs) ranked by means of a weighting expression derived from linear discriminant analysis applied to the data of Table 1

ELC	*Label*	*Weight*	*Rank*
$T1 \wedge T2 \wedge T3 \wedge T4$	*L*1	1.088	1
$\overline{T1} \wedge T2 \wedge T3 \wedge T4$	*L*2	0.854	2
$T1 \wedge T2 \wedge \overline{T3} \wedge T4$	*L*3	0.846	3
$T1 \wedge T2 \wedge T3 \wedge \overline{T4}$	*L*4	0.805	4
$T1 \wedge \overline{T2} \wedge T3 \wedge T4$	*L*5	0.759	5
$\overline{T1} \wedge T2 \wedge \overline{T3} \wedge T4$	*L*6	0.612	6
$\overline{T1} \wedge T2 \wedge T3 \wedge \overline{T4}$	*L*7	0.571	7
$T1 \wedge T2 \wedge \overline{T3} \wedge \overline{T4}$	*L*8	0.563	8
$\overline{T1} \wedge \overline{T2} \wedge T3 \wedge T4$	*L*9	0.525	9
$T1 \wedge \overline{T2} \wedge \overline{T3} \wedge T4$	*L*10	0.517	10
$T1 \wedge \overline{T2} \wedge T3 \wedge \overline{T4}$	*L*11	0.476	11
$\overline{T1} \wedge T2 \wedge \overline{T3} \wedge \overline{T4}$	*L*12	0.329	12
$\overline{T1} \wedge \overline{T2} \wedge \overline{T3} \wedge T4$	*L*13	0.283	13
$\overline{T1} \wedge \overline{T2} \wedge T3 \wedge \overline{T4}$	*L*14	0.242	14
$T1 \wedge \overline{T2} \wedge \overline{T3} \wedge \overline{T4}$	*L*15	0.234	15
$\overline{T1} \wedge \overline{T2} \wedge \overline{T3} \wedge \overline{T4}$	*L*16	0	16

Thus the front end will input to the RS the logical expressions we have labelled ***L1***, ***L2***, ***L3***, etc. in just that order, assuming that the records retrieved by each of the ***Li*** are concatenated at each stage. (If not, the input sequence would be: ***L1***, ***L1*** ∨ ***L2***, ***L1*** ∨ ***L2*** ∨ ***L3***, etc.) The process continues until some stopping criterion is met.

End of example

Lastly we note, by way of comparison, that van Rijsbergen has also suggested the use of a (non-linear) discriminating function that takes attribute dependencies into account. His method also involves estimates of weights of individual attributes (the weights for each being a function of conditional probabilities as between that attribute and an attribute adjacent to it in what is termed the 'maximum spanning tree'). The maximum spanning tree can be computed on the attribute × record data for a retrieved set. The advantage of his technique may be that it treats the problems of attribute identification and attribute weighting jointly, but a possible disadvantage may be slowness in computing the maximum spanning tree. Only through experimental work, of course, can the rate and extent of convergence of successive retrieved sets to the set of relevant documents be determined for such a method or for the method suggested in this paper.

5. LINKAGE BETWEEN AUTOMATIC SYNTAX FEATURE AND HEURISTIC FEATURE OF FRONT END, AND CONCLUSION

The FES we have described has two distinct features.

1. It allows the user first to 'seed' the retrieval system with an unstructured set of record attributes plus a numerical stopping criterion (supplied by him). The logical operations are undertaken for him.
2. The FES will prompt the user to flag records so retrieved as 'relevant' or 'not relevant' to the information need that led to usage of the system.

On the basis of such record-flagging, the FES does two things: it (optionally) re-identifies a small set of record attributes, and it constructs and orders a set of ELCs that is likely to yield a more effective information channel. The latter (i.e., the heuristic) procedure is itself iterative. The FES itself can form part of the 'host' retrieval program or it can be remote from it and local to the user, e.g., it can be stored in an intelligent terminal, or in a mini through which an ordinary terminal communicates to a remote system.

It is suggested that conventional retrieval systems (i.e., RS systems) should, to overcome initial user resistance to an unexpected form of dialogue, simply introduce the front end as an optional program path. The default path would be the RS. The author's opinion, based on some eight years of teaching online retrieval from bibliographical record systems to first-time users, is that the FES would be highly acceptable to users. Almost invariably, a first-time user obtaining an ineffective first retrieved set will turn to the instructor and comment to the effect: 'I have put in some terms and logical operators but it hasn't delivered the right references. What do I do next?' The customary reply is surely that one can return to the thesaurus, or inspect attributes in relevant-retrieved records in an intuitive way, or look for records by likely authors, or simply 'think again'. But none of these options are convincing, they are more or less laborious, and they do not exploit the power of

modern computing equipment. With an FES to switch on, the user can surely answer his own questions rather more convincingly and quickly, with the likelihood of much more effective retrieval.

ACKNOWLEDGEMENTS

I would like to acknowledge and thank Dr M. A. Heather of the Polytechnic's School of Law for drawing my attention to the existence of deontic logic; and the Editor of *Information Technology* for drawing attention to the relevant work of Porter and Dosckocs.

REFERENCES

Al-Hibri, A. (1978) *Deontic Logic.* Washington, University Press of North America.

Attar, R. and Fraenkel, A. S. (1981) Experiments in local metrical feedback in full-text retrieval systems. *Information Processing and Management 17*, 115–126.

Brookes, B. C. (1968) The measures of information retrieval effectiveness proposed by Swets. *Journal of Documentation 24*, 41–54.

Dosckocs, T. E. and Rapp, B. A. (1979) Searching Medline in English: a prototype user interface. *Proceedings of the ASIS 16*, 131–139.

Gilbert, E. S. (1968) On discrimination using qualitative variables. *Journal of the American Statistical Association 63*, 1399–1412.

Harper, D. J. (1980) *Reference feedback in document retrieval.* Ph.D. thesis. Computer Laboratory, University of Cambridge.

Heine, M. H. (1981) *Extension and application of Swet's theory of information retrieved.* Ph.D. thesis. Computing Laboratory, University of Newcastle upon Tyne.

Hoel, P. M. (1971) *Introduction to Mathematical Statistics.* 4th ed. pp. 181–186. London: J. Wiley.

Kochen, M. (1962) Adaptive mechanisms in digital 'concept' processing. *Proceedings of the Joint Automatic Control Conference* (American Institute of Electrical Engineers), 49–59.

Lipski, W. Jr. and Marek, W. (1979) Information systems: on queries involving cardinalities. *Information Systems* (GB) *4*, 241–246.

Porter, M. F. (1982) Implementing a probabilistic information retrieval system. *Information Technology 1*, 131–156.

Rocchio, J. J. (1966) *Document retrieval systems—optimization and evaluation.* Ph.D. thesis. Harvard Computation Laboratory, Harvard University.

Rocchio, J. J. and Salton, G. (1965) Information search optimization and iterative retrieval techniques. *AFIPS Fall Joint Computer Conference Proceedings 27*, 293–305.

Salton, G. (1975) *Dynamic Information and Library Processing.* Englewood Cliffs, N.J.: Prentice Hall.

Van Rijsbergen, C. J. (1977) A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation 33*, 106–119.

Van Rijsbergen, C. J., Robertson, S. E. and Porter, M. F. (1980) *New models in probabilistic information retrieval.* Computer Laboratory, University of Cambridge (British Library R & D Report No. 5587).

Vernimb, C. (1977) Automatic query adjustment in document retrieval. *Information Processing and Management 13*, 339–353.

Yu, C. T., Luk, W. S. and Cheung, T. Y. (1976) A statistical model for relevance feedback in information retrieval. *Journal of the ACM 23*, 273–286.