## SECTION B : Experiments

### I Preface

In this section the retrieval system factors we have investigated, and the retrieval tests related to them, are presented in detail. The factors are grouped under the three headings of input, indexing and output factors, and retrieval results are indentified and compared, as appropriate, under these heads, primarily in terms of recall and precision performance. The object of the section is simply to describe the tests: the results are analysed in Section C.

The specific ways of presenting performance we have adopted were considered above in Chapter A III.3, and are summarised in Figure AIII.3. The organisation of the Run Tables giving our detailed test results according to the various performance measures, as they will be referred to in the rest of this section, is indicated in the next paragraph. This is followed by a note on the notation we have adopted for characterising relative performance, and a reference summary of the factors examined in our tests.

The presentation of the individual test results in tabular rather than graph form is for greater accuracy and to save space. But as the tables are rather tedious to use, selected comparisons are illustrated by conventional recall/precision graphs, as indicated in the text.

#### Organisation of the Run Tables

A run refers to a search of a document file, by a request file, for specific request and document characterisations and searching and matching procedures. These tables therefore present the output of runs in the natural or induced formats described in Chapter III of Section A, processed to provide the different views of performance based on recall and precision discussed there. A run set refers to parallel runs on different collections.

The main run tables (M) cover runs with recall and precision based characterisation based on document cutoff, and specifically that obtained with procedure B1 of Figure AIII.3, i.e. with averaging by numbers across matching values for partially ordered output resulting from real or notional coordination level searching, followed by linear interpolation for precision at ten standard recall levels. These tables cover the investigation of input and indexing factors, and those output factor options naturally leading to this form of output. The secondary run tables (S) cover the other forms of performance characterisation, which have generally been less extensively used in the project work: though some are naturally associated with particular experiments and especially output factor tests, their main use has been to provide some alternative views of performance as a check on the procedure of the main tables.

The secondary run tables are labelled with the appropriate performance characterisation abbreviation from Figure AIII.3. The most important secondary table is that for recall cutoff procedure B4 of Figure AIII.3, giving precision at standard recall derived from fully and completely ordered output, with averaging over the precision values obtained for the individual requests by pessimistic interpolation. This is Table Src. The other secondary tables give the results for the remaining forms of recall

and precision characterisation, and for the simple numerical methods of characterisation. Document cutoff averaging by numbers across matching ranks (B2 in Figure AIII.3) appears in Table Sdr; performance using average of numbers at specific rank positions (B3) in Table Spr; and cumulative effectiveness, E, (B5) in Table Sce. The numerical characterisation in terms of average total documents retrieved (A1) in Table Str; of average numbers of relevant documents retrieved at specific rank positions (A2) in Table Srr; of average expected search length for a single relevant document (A3) in Table Ses; and of the cumulative proportion of requests retrieving their first relevant document by specified rank positions (A4) in Table Scr. As shown in Figure AIII.3, most of the secondary table procedures are associated with full and complete search output. This may have itself been generated by different methods, fully discussed in Chapter B IV below, and where such alternatives exist they are indicated in the tables, labelled dissimilarity (dis), cosine (cos), and coordination (sum). (Dissimilarity and cosine scoring coefficients naturally generate fully or almost fully ordered output, so little modification of it is required to obtain the output type required for comparisons; coordination matching typically generates partially ordered output which is converted to the fully ordered type by random ordering of tied documents). Unfortunately the effort involved in providing all the alternative performance characterisations is considerable, quite apart from reservations about their propriety in some cases. Some characterisations have therefore been omitted, for purely practical reasons. Where this applies to some members only of the range for a given run, secondary table entries are marked n.a. (not available).

The main tables giving actual performance are prefaced by summary tables (MS) indicating which runs have been carried out. The secondary tables are similarly prefaced by summary tables (SS) showing which performance alternatives and output accompany main runs. For full details see the Note prefacing the Run Tables.

The general principle of organisation adopted for the tables has been that each run should appear only once. Runs of a similar character are grouped together in a table comparing performance across collections. As a run represents a particular combination of variable values under the input, indexing and output factor headings, it can be considered in relation to other runs from different points of view. The grouping of the runs is for convenience only, since it reflects just one view of each run. The runs are grouped, very crudely, under headings representing topics of interest, but it must be emphasised that there is nothing canonical about the groupings: the runs can be categorised in other ways, and in the text runs from different tables are selected for comparison. The headings used are Terms (T), Document Description (D), Vocabulary Modification (V), Term Classification (C), Term Weighting (W), and Term Relevance Weighting (R).

In each table, the columns represent collections, main or subsidiary. Further, since for many of the collections regular alternative requests like those representing manual indexing as opposed to automatic indexing have been used, these are regarded as generating collections and so define columns in the tables, labelled a or m as appropriate. The rows in the tables represent different treatments of system components, for example, various approaches to and uses of term weighting in table W, to modifying the term lists constituting requests in table T, and so on. When runs involve relevance variants these provide a change of context

which is indicated in the tables by subheadings. For some runs, for example those involving term classification, further similar contextual changes in requests or indexing vocabulary apply, which are indicated by additional subheads.

The same layout is adopted for both main and secondary tables. Within each table the run sets are simply numbered in sequence, with corresponding numbers for main and secondary tables. The labelling system for runs thus consists of a M or S referring to main or secondary tables, with modifiers tr, rr, ... etc. for the latter, T, D, V ... etc. referring to the specific table, and 1,2,3 ... etc. for the run set within the table.* Different collections are referred to by their standard name, followed if necessary by their request form letter. We thus have, for example, MT1 for the baseline term coordination run set, with C200I-m and K400I-m as comparable collections with manual request indexing or K400A-a and K400A-m representing alternative requests for the same collection; while ST1 provides different outputs and performance characterisations for, say, the C200I and K800T collections. Again MR2, MR10 and MR13 refer to run sets representing different weighting tests involving, for instance, the C200I and K400I collections. Note that if to save space empty sections of tables are left out, in the secondary tables, the run set numbering is consistently maintained with a run having the same reference number across the alternative performance representations.

Some miscellaneous runs, in particular those for Boolean search specifications for the U27000P collection, do not fit naturally into the tables just described. They are therefore lumped together in a separate other Runs Table (O).

Notation

For summarising sets of results, a notation has been adopted as follows:

x = y the results are the same
x >= y x is sometimes better than y, sometimes the same; abbreviated as x is slightly superior to y
(x <= y x is slightly inferior to y)
x > y x is superior to y
(x < y x is inferior to y)
x >> y x is much superior to y
(x << y x is much inferior to y).

When the attempt is made to draw an overall conclusion from a set of results naturally exhibiting variable differences, a precise characterisation of relative performance is impossible, and individual results may not satisfy the requirement for statistical significance mentioned earlier. If consistent and similar performance differences are not manifest, summarising a set of results can only indicate a tendency. In general, while x = y means that results generally do not show significant differences, x > y (or x < y) means that there is generally a statistically significant difference, and indeed one which can be characterised as noticeable (see Chapter III.3 of Section A); and x >> y (or x << y) means that there is a material significant difference. With x >= y (x <= y) some results will be statistically, and noticeably, different, others not. Note that when results are contradictory, with x sometimes superior to, and at other times inferior to, y, a summary conclusion is not attempted. When two specific runs are being compared, the notation x >= y means that precision performance is different at some recall

* If the modifiers are omitted reference is to all the secondary tables.

levels, but the same at others.

In general, when comparisons across a set of results justify a summary conclusion, this will be presented using the notation; when only a few runs , or a single collection, are involved, or when results exhibit discrepancies, the results will be given verbally: thus the occurrence of notational expressions in the text is usually a clue to more important tests.

Summary of factors tested

The variables and value sets we have been concerned with in our tests were listed in Figure AIII.2. However, since individual runs necessarily involve some choice under all three factors, the variables are again listed for reference below. This will hopefully mean that when runs are considered from a particular point of view, in a particular context, any references to their other features will be sufficiently comprehensible; or at any rate that the reader will have some feeling for what other variables have been assigned values. The detailed treatment of the variables appears under the factor headings in the chapters which follow.

| | |
|---|---|
| Input factors | : indexing mode |
| | indexing source |
| | description exhaustivity |
| | vocabulary specificity |
| Indexing factors | : term classification type and use |
| | term weighting type and use |
| Output factors | : scanning strategy |
| | matching condition |
| | scoring criterion |

## II Input Factors

### 1 Inputs to Indexing

The questions to be investigated here concern the effects of four major factors on subsequent indexing and search performance, i.e. we are concerned with the properties of the basic data of indexing with respect to documents and requests. Some points apply to both documents and requests, but there are important differences between the two, and they will therefore be discussed separately.

The systematic investigation of input factors has presented most problems for the project, and we fear that the work we have done on them is both apparently and actually partial and ad hoc. Some difficulties arise from the interdependence of the variables involved, which can only be studied in extensive series of experiments outside the scope of all but the best funded projects; but these difficulties have been compounded in our case by the inadequacies (from our point of view) of data obtained from elsewhere, over the generation of which we had no control.

Though our experiments with input factors have been primarily designed to throw light on the effects of properties of the input material on the behaviour of retrieval systems incorporating the indexing devices in which we have been mainly interested, namely term weights and classes, they can also be looked at in another way: the tests can be used simply to determine what forms of input are most effective in general. We believe that some conclusions can be drawn from our experiments about both specific and general requirements of input, but the limitations of our data mean that they must be somewhat tentative.

The basic data for our experiments is provided by the primary indexing of the documents and requests: as described in Chapter A II, this consists of extracted keyword stems defined prior to searching for most of the test collections, but at the time of search for the U27000P collection. This primary indexing in fact represents the first step in the processing of indexing viewed as a sequence of operations applied to given data, keywords, to obtain index keys representing documents or requests (the former in particular) for searching. The extraction of the keywords itself is strictly the first step, but since we have dealt almost entirely in our project with index descriptions of the simple postcoordinate kind, we have taken the derived word list rather than its source text as the starting point of processing.

We may then view the word list as a tentative index description, which is subject to revision in that items are removed or added, or their relative status altered. An important question about revision processes is whether elements of the initial lists are eliminated without trace or whether they can be recovered, as in weighting schemes which temporarily suppress items by giving them zero weight. Word grouping schemes leading to higher level or alternative descriptors (stems, thesaurus terms, or phrases) then represent a compromise in that the specific input words are lost, but information associated with them is retained. The request based indexing of the U27000P collection reflects the extreme view that no processing should be applied to documents before searching in case anything is lost, while the request specification with its particular choice of truncation, phrase etc. embodies processing supposedly most appropriate to the user's requirements.

The distinction between input and indexing factors made in Section A refers to an essentially arbitrary cut across the possible sequence of input processing, i.e. indexing steps: it is based on the simple two step case where words are first extracted and then replaced as final index keys by e.g. word groups functioning as higher level descriptors. The input factors listed were regarded as relating to natural properties of the extracted lists. However, it is evident that if the natural properties of the lists are deemed unsatisfactory for retrieval, the lists may be modified. Thus an automatically extracted document word list may be consciously altered with results similar to those following intuitive decisions in human word selection. The modification processes might more properly be assigned to indexing: after all grouping as an indexing operation is a response to natural properties of extracted word lists. Indexing source and mode are perhaps the only genuine input factors of those listed; but since description exhaustivity and vocabulary specificity are not necessarily or wholly subject to manipulation, they are treated as input factors in this report.

Further, since there is no evidence that stemming is unhelpful, whether applied directly to documents or via requests, we have taken the extracted word stem, or term, lists constituting the primary indexing for all our collections except U27000P as the basic data or raw input for indexing; i.e. we disregard the logical status of stemming as an indexing operation. Experiments by Salton (1968a, b) and Cleverdon (1966) show that stems never perform worse than word forms, and sometimes perform better ; and a comparison between C200A and C200Aw collections. representing stems and word forms respectively, for runs MT1, shows no superiority in the latter. An additional practical reason for using stems is the reduction in the size of the indexing vocabulary.*

Searching on the primary indexing gives the performance represented by the run sets MT1 and ST1. Differences in performance for different collections are considered in more detail below. For individual collections, as mentioned in Section A, the level of performance represented by the baselines in particular is used as a standard reference point. Where appropriate it is indicated by a dotted line on illustrative graphs.

In the next sections we examine the effects of features of this basic data on retrieval performance under the four heads of indexing mode, indexing source, description exhaustivity and vocabulary specificity. It is unfortunately virtually impossible to study these in a totally independent way: exhaustivity and specificity are logically related, and with our limited data choices of indexing source and indexing exhaustivity cannot be treated separately. However, our tests suggest some desirable properties of the final data to which weighting and classification may be applied, sometimes associated with the raw input and sometimes resulting from modification of this data in a first stage of processing.

1.1 Indexing Mode : manual or automatic

As noted in the Introduction, the overall object of the project has been evaluation of fully or partially automated indexing techniques. We are concerned here with the specific comparison between automatic and manual

* The fact that more or less extenstive truncation is widely practised in operational systems (usually through requests) suggests it is useful though there have been few controlled experiments to establish exactly how useful it is.

extraction of the basic data. Early work on automatic indexing was concerned with wholly automatic document and request processing, including the extraction of words or more complex items, especially from full texts, by statistical or linguistic methods (see Sparck Jones 1974). Partially automated indexing may involve the automatic processing of manually obtained data, or human editing of automatically obtained material. Fully automated indexing is an appealing idea, and the attempt to demonstrate that it can compete with human indexing, primarily in terms of effectiveness, but also in economic efficiency, has occupied some research workers in the field for years. Salton in particular has claimed that automatic indexing (and searching) is competitive with manual. (Salton 1972, 1975a).

The research on automatic term classification preceding the present project was originally intended to show that automatic techniques could provide what was then supposed to be the most influential component of a retrieval system, namely the indexing language; but while this should in principle be derived and used in a fully automatic way involving automatic term extraction, grouping and description, it was suggested that a good compromise would be the automatic improvement of index data initially obtained manually, in a very simple way, say by listing salient abstract words. Simple manual word selection might provide better data, more cheaply, than automatic selection. The classification tests with the Cranfield material reported in Sparck Jones 1971a were on this basis. The present project suggests, as will be discussed in more detail later, that partially automated indexing involving initial manual extraction followed by automatic processing e.g. to generate weights appears competitive in performance and practically convenient. However, the situation is complicated by the fact that there are both documents and requests to be processed, and the choice of automatic or manual processing may not be as well suited to the one as the other.

Properly, comparisons between manual and automatic extraction should be based on the same source, and should involve term lists of comparable exhaustivity. We have been able to restrict our comparisons in this way only in a few cases.

Comparisons providing some information about the relative merits of automatic and manual word extraction are as follows.

Concentrating on documents, we first consider primary indexing performance irrespective of variation in source and exhaustivity, for the Cranfield and Keen collections. Manual extraction is represented by I collections, automatic by A and T. Run set MT1 gives performance for the regular relevance sets, covering searches by manually and by automatically indexed requests. Run set MT2 gives results for the high relevance variants, and run set MT3 for the B variants for the Cranfield 200 collections. The runs taken together show abstracts inferior to titles and the latter equal to manual indexing for the Cranfield collections, with abstracts equal to titles usually less good than manual indexing for Keen. Overall, using the notational convention introduced earlier, this suggests the conclusion A <= T <= I.

For indexing from a common source, though to different exhaustivity, we compare K400I and K400A. The relevant results in run set MT1 show A <= I. For comparable exhaustivity, though from different sources, K400T and K400I with both manual and automatic requests, and K800I for the former, provide results in run set. These show T < I. We have not been able to make any simple comparisons with the same source and exhaustivity,

since we have not been qualified to enlarge our test data by providing additional manual indexing.

We have, however, extended our range of comparisons related to similar exhaustivity by artificial techniques for modifying the initial document descriptions. One is to process the term lists derived from abstracts by deleting terms occurring only once within the abstracts, on the presumption that they are less significant. This gives an average of 16.7 terms per abstract for the C200A collection and of 12.7 for the K400A collection (see Figure BII.1), making abstracts comparable with titles for Cranfield and with both titles and manual indexing for Keen. The results for the relevant searches appear in run sets MT1 and MD1. These show abstracts inferior to titles for Cranfield, and comparable to titles but inferior to indexing for Keen. A second basis for comparison for the C200I collection only is provided by the use of descriptions from which terms have been randomly deleted (see Sparck Jones 1973b), giving an average of 16.4 terms per document. Runs MT1, MD1 and MD2 indicate that these rather rudely modified descriptions are inferior to both the modified abstracts and titles. Taken together, these comparisons show no consistent pattern, but this is perhaps not surprising.

The reduced Keen abstracts allow a comparison between the K400I and K400A collections involving the same source and similar exhaustivity: in this case the automatic indexing is inferior to the manual.

Comparisons between automatic and manual request processing are much simpler since the source is the same, and exhaustivity is not very variable. The tests are those involving the two types of request of the Cranfield and Keen collections. When searched on abstracts the automatic requests are somewhat inferior to manual, on titles they are the same. The overall conclusion is thus that a <= m. In these tests the manual indexing consists simply of word selection. For the Keen collections 'good' requests with some added terms are available. As run set MT1 shows, these perform slightly better than the simple manual requests for the K800I and K400I and K400I, A and T collections; i <= g. The UKCIS Boolean profiles can be treated (as will be discussed in more detail later) as simple term lists: these are very different in size from the automatic ones. A slightly unfair comparison between these two, i.e. runs MT1 for the U27000T and U27000P collections respectively show great superiority in the amplified profiles.

All these comparisons depend on the primary indexing or on variations with simple term matching. As a cross check comparisons across runs with another major variable change are required, and suitable parallel runs though not quite such extensive ones, are provided by those involving term weighting using term collection frequency information see Chapter B III.3 below. The relevant results appear in run sets MW7 and MW8, and show a similar pattern to the previous comparisons. For documents irrespective of source and exhaustivity variation manual indexing tends to perform better, the general picture being A <= T <= I. For comparable sources the Keen 400 collections show abstracts equal to manual indexing and for comparable exhaustivity the Keen 800 and 400 collections show titles inferior to indexing, T< I. The comparisons for the two forms of request again show automatic equal to manual, a = m, and for the K800I and K400I collections the good requests perform slightly better than the plain manual ones, i <= g.

These cross checks reflect changes in indexing factors, specifically

the adoption of one indexing device, weighting, instead of another (the null one of primary indexing). Cross checks should also be made allowing variations in output factors. As will become clear in the full discussion of output factors in Section B, project tests involving changes of major output variable have been few, while those involving changes of sub variable are of limited value as cross checks, and are perhaps better considered in the context of the detailed analysis of output factors. It may therefore simply be noted that the indexing cross checks using collection frequencies do also involve a change of minor output variable.

It is more important to show how the comparisons are affected by the different techniques for evaluating performance described in Chapter III of Section A.

Turning now to the picture presented by the alternative methods of performance represented by the secondary tables, before commenting on the specific results for indexing mode, some general points about the secondary characterisations should be noted. These refer to all the methods except tr, total documents retrieved, which are all derived from fully and completely ranked output. First the scoring coefficients dis and cos involve an element of normalisation which is absent from sum. The latter has been used chiefly for relevance weighting, for which the former appear inappropriate, and sum runs are therefore considered primarily under this heading, and in the direct comparisons between dis and cos on the one hand and the sum on the other in relation to output factors. However, it may be noted that where results from sum are available, as for indexing mode, comparative findings for it parallel those for the other coefficients. The difference between dis and cos is in practice trivial, and duplicating runs for both would be ridiculous as well as economically ruinous. For historical reasons dis has been used in some cases and cos in others, but it may reasonably be assumed that either is representative of both.

Secondly, these methods are all related in other various ways which should be borne in mind. The main alternative performance characterisation by recall cutoff, rc, and also cumulative effectiveness, ce, expected search length, es, and cumulative requests, cr, are derived from individual request results. Averaging by document rank, dr, precision and recall at specific ranks, pr, and average relevant retrieved by specific ranks, rr, on the other hand are all associated with averaging by numbers across ranks, i.e. with pooled query results. However, overall differences in the views of performance given by the different methods are noticeable. For example, rr tends, hardly surprisingly, to show very little difference over runs for the same collection. Again, though the different methods tend to give similar results overall for particular experimental comparisons, the simple measures like es and cr, and the former in particular, may differ from the general picture.

As method tr, total documents retrieved, is associated with the output used in the main tables, information is given for far more runs than in the other secondary tables. The tr figures are indeed intended chiefly as an information supplement to the main tables, and will be referred to specifically only for those runs also covered by the regular secondary tables. It should be noted that some comparisons cannot be made with tr results, for example, between those for term matching and for collection frequency weighted term matching: any performance differences here are reflected in the output ordering and not in the totals of documents retrieved, which are the same.

The final point to be made about the secondary performance characterisations is that as the detailed discussion of each for every factor studied would be oppressive, only the general conclusions to be drawn from the set of results for each factor are presented. These are necessarily couched in rather vague language, and specific references to noticeable or material differences are only occasionally possible. Indeed a general view is what is really required: the reader is referred to the tables for the detailed results. It should be noticed that as the secondary tables cover far fewer runs than the main tables, in drawing general conclusions from them about particular factors, variations in other factors may be disregarded.

Considering now the secondary results for indexing mode, the relevant runs are those for ordinary term matching, ST1, including both manual and automatic requests, and those for collection frequency weighting, SW7. The results here support the main ones, as they indicate a tendency for manual document indexing to perform better than automatic indexing; however, there is no real difference between automatic and manual requests for the relevant abstract and title collections. The tr results are in general accord with the others; in particular automatic indexing of abstracts tends to be inferior in retrieving far more non-relevant documents for little or no gain in relevant ones.

An overall, but necessarily summary, conclusion on indexing mode is that automatic extraction tends to be inferior to manual. The reasons will be considered in Section C.

To illustrate the effects of document indexing mode, performance for the C1400m, C200, K800 and K400 collections is shown in Figure BII.2, comparing automatic and manual indexing in relation to text, abstracts and titles. These and other illustrations below are, except where specifically mentioned, all for the regular coordination matching of the main tables.

1.2 Indexing Source : title or abstract or text

In the early retrieval literature, a good deal of attention was devoted to the respective merits of different indexing sources, i.e. to comparisons between titles, (titles +) abstracts, and (titles + abstracts +) full texts. For example Salton 1968a,b concluded that titles were inferior to abstracts, while texts were not significantly or at any rate usefully superior to abstracts. Experiments in fully automatic indexing by, for example, Damerau 1965 and Dennis 1965, 1976, were based on full texts; but the effort of working with full texts is considerable, and since there is no evidence that retrieval performance is materially improved by text based indexing, there has seemed to be nothing to lose and much practically to gain from working with abstracts or even titles. Many operational systems are confined to titles, perhaps supplemented by a few abstract derived keywords.

As with the tests on indexing mode, the limitations of our data have made it impossible for us to make proper comparisons between titles, abstracts and texts. Thus for real comparisons between different sources, the indexing should be in the same mode, and should provide descriptions of similar exhaustivity. The latter implies selectivity, particularly in automatic indexing based on texts or abstracts, which we have not generally been able to attempt.

The limited comparisons possible essentially represent alternative views and groupings of the runs considered in the previous section. But these comparisons are confined to manually indexed requests, both for simplicity and because the automatically indexed requests did not differ noticeably in performance. The comparisons of course concern the documents only.

The initial comparisons are for the primary indexing, disregarding differences of mode and exhaustivity. They apply to the Cranfield and Keen collections and compare titles T, abstracts A and texts I for the Cranfield 1400 and 200 collections; titles and abstracts, used forthe manual I indexing, for Keen 800; and titles and abstracts, the latter used for both manual I and automatic A indexing, for Keen 400. The relevant results are in run set MT1 for the regular relevance sets, with relevance variants in run sets MT2 and MT3. The runs show differences between the Cranfield and Keen material, with abstracts inferior to title and title the same as text for Cranfield, but with titles slightly inferior to abstracts for Keen. When comparisons are restricted to the same indexing mode, i.e. to A and T for Cranfield 1400 and 200 and Keen 400, the same contrast applies, with abstracts slightly inferior to titles for Cranfield, and titles slightly inferior for Keen. No comparisons with similar exhaustivity as well as the same mode are possible for the primary indexing.

The modification of the original descriptions to achieve comparable exhaustivity described above throws some light on the value of the different sources. If differences of mode are disregarded, when the abstracts are purged of terms occurring only once within them, abstracts A for the Cranfield 1400 and 200 data can be compared with titles T, while the abstract derived indexing A and I can be compared with titles T for Keen 400. As run sets MT1 and MD1 show, abstracts <= titles. With the randomly reduced manual index descriptions I for Cranfield 200, runs MT1, MD1 and MD2 show the modified texts inferior to either the original or modified abstracts, with the latter much inferior to the titles.

The purged abstracts also allow comparisons for the same mode and similar exhaustivity for the Cranfiled 1400 and 200 and Keen 400 collections. Run sets MT1 and MD1 show the modified abstracts <= titles.

As a cross check to see whether the primary indexing differences of performance are maintained when another system variable value is altered and specifically, as for indexing mode, when a device generally improving performance is adopted, some limited comparisons involving collection frequency term weighting can be made. When mode and exhaustivity are allowed to differ, run sets MW7 and MW8 show abstracts slightly inferior to titles, the latter equal to titles for Cranfield 1400 and 200, while titles are inferior to abstracts for Keen 800 and 400. That is , the results parallel those for the primary indexing. When the comparison is confined by indexing mode, so abstracts A and titles T are compared, Cranfield 1400 and 200 show abstracts slightly inferior to titles, Keen 400 titles inferior to abstracts. No more extensive comparisons are available.

The alternative performance characterisation of source results in the secondary tables again covers term matching run sets ST1 and collection frequency weighting SW7. The figures support the conclusions to be drawn from the main comparisons, namely that perhaps full text is preferable to

either abstracts or titles, though this finding may be due to differences of indexing mode or exhaustivity; but that there are no clear advantages for abstracts or titles as the Cranfield and Keen collections behave differently. The tr figures, as in the previous section, show automatic abstracts inferior.

Overall, no summary conclusion is possible indicating the relative merits of the different indexing sources, because the two bodies of material permitting comparisons, namely Cranfield and Keen, generate collections with contrasting behaviour. Explanations for the results will be attempted in Section C; for the moment it may simply be noted that there is no consistent superiority of one source over another.

The graphs of Figure BII.2 for the various collections, used for the previous section, can also be used to illustrate the discussion of indexing sources.

1.3 Description Exhaustivity : low or medium or high

This again is a topic which has already been investigated in some detail: Cleverdon has suggested, for example, that there is an optimal level of indexing exhaustivity for a given set of documents (Cleverdon 1970). Unfortunately it is not clear how this is to be determined in advance, especially for a growing set of documents; and it is also not clear how critical the level is. Sparck Jones 1973b argued that the level is not very critical, and further that documents and requests can complement one another with respect to exhaustivity in a quite effective way, in particular to overcome a 'mistaken' choice of level for document indexing.

While exhaustivity strictly refers to document or request description as used for searching, with automatic indexing techniques of the kind studied by the present project the exhaustivity of the primary indexing will parallel that of the final descriptions and hence may affect their performance. The tests considered in this section are therefore intended to provide information about the relative effects on retrieval performance of different levels of input description exhaustivity.

For proper comparisons descriptions of different exhaustivity should be obtained from the same indexing source by indexing in one mode. Such alternative sets of descriptions could of course hardly be obtained from titles, but could derive from abstracts or full texts. We were not in a position to carry out manual indexing to different levels (as was done in the Cranfield 2 project), and have not attempted to apply sophisticated automatic indexing devices designed to select fewer terms from, say, abstracts than are ordinarily obtained, when only stop words and morphological variants are eliminated.

In the same rather unsatisfactory way as for the previous factors, we are obliged to treat the primary indexing descriptions for collections derived from the same data as substantially comparable, in this case with respect to exhaustivity, though they are also characterised by differences of mode or source. Figure AII.4 shows that titles generate less than 10 terms, while abstracts exceed 50, with manual indexing ranging from about 7 for Keen to about 30 for Cranfield. For convenience, in relation to our collections, we may characterise document descriptions having 1 - 14 terms as of low exhaustivity, or short; 15 - 34 as medium; and above 35 as of high

exhaustivity, or long. For requests, 1 - 5 may be deemed short, 6 - 11 medium and over 12 long.

Initial comparisons for documents are for the primary indexing, disregarding differences of mode and source. They cover short, medium and long descriptions for the Cranfield 1400 and 200, representing titles, manual indexing and abstracts respectively, and short and long for Keen 400, the former representing both titles and indexing, the latter abstracts. Again considering only runs for manually indexed requests, run sets MT1, MT2 and MT3 give performance for the regular relevance judgements and for relevance variants. Overall the results show long <= medium <= short. Comparisons restricted to descriptions obtained in the same indexing mode refer to the Cranfield 1400 and 200 automatic indexing giving long abstracts and short titles. Run sets MT1, MT2 and MT3 for the relevant collections show long <= short. Indexing from the same source, in different modes, is represented by the Keen 400 long abstracts and short manual indexing. Runs MT1 and MT2 show the former slightly inferior to the latter. There is no primary data supplying indexing at different levels of exhaustivity derived in the same mode from the same source.

In investigating exhaustivity, modification of initial descriptions must be regarded not as a means of making different descriptions comparable, as in the previous sections, but as a means of generating a range of different descriptions. Of course any indexing devices may change description exhaustivity, more or less obviously, and may be intended to do so. For example, the addition of class related terms, or the use of weights, which effectively increase or reduce the incidence of terms, alter exhaustivity; changes to the indexing vocabulary as a whole will incidentally affect individual descriptions. But the effects of such operations may be more complex than those following naturally from the simple provision of more or less terms for each input document individually. The implications of the various indexing devices for exhaustivity will therefore be considered as appropriate later.

In Sparck Jones 1973 rather heterogeneous procedures were used to modify given descriptions, perhaps changing too many system variable values at once. More limited and so hopefully more reliable comparisons are restricted to the following modifications.

The modification of abstracts by removing terms with an interanl frequency of 1 permits a comparison between long and medium descriptions for the C1400A, C200A and K400A collections. The conclusion to be drawn from run sets MT1 and MD1 is that medium = long. These cases can legitimately be regarded as reflecting primary automatic indexing to different levels of exhaustivity from the same source. A looser comparison for the C200I collection is before and after random deletion of terms: a rather drastic reduction in description length to half the original leads to inferior performance, as shown in runs MT1 and MD2. (Lesser reduction does not degrade performance much.)

Since the initial verbal statements of requests may be quite long, as some were for the Keen data, different levels of request indexing exhaustivity are possible. But we have not had much opportunity to study them. Automatic requests tend to be somewhat longer than their manual counterparts, but only for the Keen data is the difference material, with the automatic requests for the K400A and K400T collections over twice as long as the manual ones. Comparing performance for these, with the automatic requests as medium length and the manual ones as short, gives medium <= short

(run sets MT1 and MT2). Few comparisons restricted to request indexing in the same mode are possible. The good requests for the Keen collections are somewhat longer than the ordinary manual requests, though the difference is small. Run set MT1 shows that for the K800I and K400I collectons the shorter requests are slightly inferior to the longer ones. Unfortunately, a comparison between the UKCIS profiles treated as simple very long term lists and the much shorter requests of the U27000T collection is not proper, since there is not merely a difference in indexing mode, but in the whole treatment of terms.

Although the collection frequency weighting scheme used to provide cross checks in the previous sections for indexing mode and source can be regarded as altering description exhaustivity, it may be expected to do this consistently, and tests involving it should therefore not be biassed with respect to description exhaustivity. Considering first differences of exhaustivity regardless of those of mode or source, we can compare Cranfield 1400 and 200, and Keen 400, with long abstracts, medium indexing and short titles for the Cranfield data, and long abstracts and short indexing or titles for Keen. Run sets MW7 and MW8 indicate a conclusion long <= medium <= short. When comparisons are confined to indexing in the same mode, for Cranfield 1400 and 200 long abstracts and short titles, the runs show long <= short. A single comparison for Keen 400 between abstracts and manual indexing from the same source runs MW7 shows the former slightly inferior to the latter. There are no comparisons with the same mode and source. For the weighting, there are no tests for the modified descriptions.

Request exhaustivity comparisons are again confined to the Keen data. Those between medium automatic requests and short manual requests, for K400A and K400T, are represented by runs MW7, which show medium slightly inferior to short, medium <= short. Comparisons between the shorter manual and longer good manual requests can be made for both the Keen 800 and 400 data. From runs MT1, for K800I and K400I, it appears that the short requests are inferior to the less short.

The secondary table results for exhaustivity are again those used previously, represented by run sets ST1 and SW7. As for the previous factors the different methods show some variation, but there appears, as for the main tests, to be a tendency for medium length descriptions to perform better, but the relative merits of long and short descriptions differ for different collections. The rather artificial and limited comparison for the C200I collection between terms and randomly reduced document descriptions (run SD2), shows the former superior. The tr figures parallel the other results.

Taken together, the tests show that in general for documents and requests higher exhaustivity is associated with lower performance, i.e. long <= medium <= short, except that for requests carefully constructed longer ones may be slightly superior to shorter ones.

Figure BII.2 provides illustrations of some effects of indexing exhaustivity. Figure BII.3, for the C1400A, C200A and K400A collections, compares regular and reduced abstracts.

1.4 Vocabulary Specificity : low or medium or high

This factor differs from the previous three in referring primarily to

documents (or requests) when taken together as a set, rather than as individuals.

It is well known that indexing vocabulary distributions conform more or less to the Zipfian one. This distribution pattern is a general one for masses of text, and is not materially affected by different choices for such input text processing factors as mode, source and exhaustivity.

The question is whether the vocabulary provided by the initial document descriptions should be modified in any way, to obtain a revised vocabulary and descriptions which may then be subject to the application of further indexing devices. If the natural distribution has detrimental effects on performance, how should these be suppressed? It is not realistic to attempt to control the use of terms, with a view to counteracting natural distribution patterns, when individual documents are initially described; it is more reasonable to consider the actual behaviour of terms and eliminate unsatisfactory ones. The problem is the criteria to be applied in doing this. When vocabularies are constructed manually it may be suggested, for example, that very general words, or very vague words, or very specific words, should not be included in the term list. In the present context the question is whether there are criteria for selecting good index terms for a set of documents which can be applied automatically, i.e. criteria related to the given pattern of term distribution.

Selecting good terms (or eliminating bad ones) is particularly important when descriptions are derived from long sources like abstracts or full texts, which naturally produce many different words. Some may be eliminated on internal grounds, say because they do not occur frequently in the given text; and others may be eliminated by reference to their general linguistic behaviour: thus a word which occurs frequently in any text would not be characteristic of a specific text. But further selection to obtain a final list of appropriate terms may still be thought desirable, and this would naturally lead to an attempt to choose terms by reference to the collection context of the document. Even when initial descriptions are short, purging the initial vocabulary may be suggested as a means of improving performance.

It has been suggested that terms which occur in many documents in a collection will not be very discriminating, i.e. will lower precision, and so should be eliminated, while those with very low frequency have no recall value, and perhaps little precision value either. Middle frequency terms are of most value, for both precision and recall. In experiments with the 200 Cranfield abstracts data, Svenonius 1972 compared the effects on performance of removing all the terms with frequencies falling in specific ranges, namely the highest, high medium, low medium and lowest quartiles of postings; and concluded that terms in the top and bottom quartiles could be removed, and indeed that the former should be eliminated to improve precision.

Removing terms on the basis of simple postings frequency is a crude procedure particularly appropriate to binary document descriptions. If information about within-document frequencies is available more refined approaches are possible. Early work by Dennis 1965, 1967 used both frequency over documents and within documents to derive an indexing vocabulary for a set of documents, and this idea has been studied by Salton and the SMART Project workers (See Salton 1972, 1973a and b, 1974, 1975 b and c). They have argued that the most effective index

terms are those which discriminate as much as characterise, and that the best discriminators are those terms which have a medium posting frequency over the collection but skew frequency with respect to the documents in which they occur. Thus if two terms a and b have the same overall postings frequency, but a has a higher frequency in the documents in which it occurs, a is more effective as a means of distinguishing one document from another. The use of a rather than b as an index term will increase the separation between documents which is a prerequisite of the selection required for useful retrieval, since typically only a few documents out of a collection are relevant to a request. Indexing with discriminators will separate documents from the body of a collection, but not from very similar documents which are presumably co-relevant.

The discrimination value is computed using a particular function Q. This represents the density of a document set as the sum of the correlations between the individual document description vectors and the centroid or average vector. An initial value of Q, $Q_o$ is computed for the given document descriptions, and alternative values, $Q_i$, are then calculated for each term i, representing the document space with this term removed from the vocabulary. If $Q_i = Q_o$ i is neither good nor bad; if $Q_i > Q_o$ this means that the documents have become more alike, so i was in fact a discriminator and should not have been removed; if $Q_i < Q_o$ the documents above become less like, so i was not a discriminator. The Q values for all the terms can be ranked and those terms with $Q_i < Q_o$ should be deleted from the indexing vocabulary; those with $Q_i = Q_o$ may be removed without performance loss to reduce the volume of data. A useful feature of the approach is that not only are terms ranked by merit, as for other functions, but a natural cutoff is supplied. (The precise formulae used are given in Appendix 1).

The obvious question about deleting terms, particularly if they are frequently occurring ones, is the possible impact on recall. Dennis found the vocabulary selected by her preferred functions plausible, but her retrieval tests were too limited for a proper evaluation of the effects of selection in performance. The SMART experiments reported in Salton 1973b and 1975b show a slight improvement in precision over the original term indexing, but the use of completely ranked output conceals any consequences of term deletion for recall.

In general, when terms are ranked by Q, the worst terms are those occurring very frequently over the collection, and specifically in many documents, which also tend not to have a variable within document distribution. The null terms (with $Q_i = Q_o$) are the very rare terms. When we calculated the function for the binary document descriptions of the C200I, I500I and IC800I collections the ranking roughly corresponded, not surprisingly, to the overall collection frequency of terms, i.e. the number of documents in which they occurred, though the precise value of a term must depend ont only on its frequency but on the particular documents in which it occurs.

Vocabulary specificity appears to be a more important factor for performance than the other input factors considered; in particular it bears heavily, as we shall see in the next section, on the indexing factors we have been concerned with, namely term weighting and classification. The experiments we have carried out relating to specificity are therefore rather more extensitve than those described so far.

The tests fall into related groups designed to study the effects on performance of removing terms from the vocabulary according to different frequency based criteria, and of deleting terms defined as unsatisfactory in specific ways. One set of comparisons removes terms falling into posting moieties, and a second those falling into quartiles; the third covers detailed studies of bad term deletion.

We initially consider therm frequency defined solely by the number of documents in which a term occurs: for many of our collections only binary descriptions are available. We first divide the vocabulary into frequent and infrequent terms, each set accounting for half the total postings. For incidental historical reasons the separation is rather crude for some collections, so only approximate posting moieties are obtained. The general pattern of vocabulary distribution means that there are only a few frequent terms, and many infrequent. Figure BII.1 gives the division point for the test collections (maximum and minimum frequency etc. are shown in Figure AII.4). Since the suggestion is that the less frequent terms are superior to the more frequent ones, deleting the frequent terms was tried on most of the collections. For the abstracts terms occurring only once were deleted as well, as the data was used for other purposes: this would have no effect on performance. Run sets MV1 and MV9 give performance for the purged primary indexing for the Cranfield, Inspec and Keen data. When the runs are taken together, irrespective of differences in mode, source or exhaustivity for the different collections, and are compared with the full primary indexing (MT1), i.e. we compare infrequent and all, we find that for those recall levels reached by both, performance is the same, but that the recall ceiling for infrequent is generally lower. For the Keen I and T collections in particular, it collapses to 20% or less. This specific pattern of result may be summarised by infrequent (=) all.

When the comparison is confined to indexing in the same mode for manual indexing I for Cranfield, Inspec and Keen, the recall loss is not too great, so infrequent = all; but automatic indexing for Cranfield and Keen, A and T, shows very poor recall for K800T and K400T, endorsing the overall conclusion infrequent (=) all. Limitation by source, for K400I and A, and by exhaustivity, for Keen I and T, does not introduce any variation. More strongly limited comparisons are not possible.

Complementary deletion of infrequent terms is clearly damaging and so has not called for extensive tests, as runs MV2 for C200I and K800I show. The value of infrequent terms is confirmed for C200I by run MV11 which give performance if some only of the infrequent terms are randomly deleted. This again is inferior to that for the full vocabulary.

When frequent and infrequent deletion is compared directly, the latter is superior to the former except in failing to reach high recall. The conclusion can be expressed by frequent (<) infrequent.

As Figure BII.1 indicates, simply dividing the vocabulary may mean, particularly for titles, that terms with quite low frequencies are deemed frequent; and since many request terms are frequent, the loss of recall is predictable. The categorisation of terms by posting quartiles, into very frequent, fairly frequent, fairly infrequent and very infrequent allows a more careful determination of useful terms. Tests deleting terms in these four quartiles were carried out on the C200I, I500I and K800I collections. The cutoff frequencies are given in Figure BII.1.

Comparisons between run sets MV3, MV4, MV5 and MV6 respectively and the primary full indexing performance of run set MT1 shows all <= minus very frequent (i.e. when very frequent terms are deleted), but all >> minus fairly frequent and all >> minus fairly infrequent, while all > minus very infrequent. Comparing the quartiles with one another can be summarised by minus fairly infrequent = minus fairly frequent <= minus very infrequent <= minus very frequent.

In these tests indexing mode was the same but source and exhaustivity varied for the different collections. The only more limited comparison available, between I500I and K800I for the same mode and source, shows no divergence from the overall pattern.

As it appears that frequent, and especially very frequent, terms are of less utility than others, and also that very infrequent ones are not helpful, a more detailed study was made of the effects of different specifications of such terms. In particular simple approaches were compared with Salton's more sophisticated Q function.

Q values were computed (following the formulae of Appendix 1) for the terms in the C200I, I500I and K800I collections, using document frequencies only. The resulting term rankings show that the bad discriminators generally coincide with those defined as frequent by posting moiety, while terms occurring only once are 'null' discriminators since removing them has little effect on Q. Deleting bad discriminators would therefore have much the same effect on performance as deleting frequent terms, as described above. A more detailed study was made using the K400A collection. To reduce computational effort a very few terms occurring extremely frequently, and also those occurring only once, were taken out of the vocabulary: this has no effect on performance compared with the full vocabulary.* Progressive reductions of the vocabulary were then represented by removal of the top quartile very frequent terms, as in the tests above, by the removal of upper moiety frequent terms, as above, and by the deletion of bad discriminators, which in this case corresponded to a slightly lower frequency threshold than that used for the moiety. Runs MV7, MV8, MV9 and MV10 gives comparative performance. Precision is the same for the recall levels reached, but the removal of frequent terms, particularly in the last case, leads to a disastrous drop in the recall ceiling.

Q values based on collection frequency only represent a somewhat unfair test of Salton's function, since it is intended to relate term behaviour over documents to behaviour in documents. Q values using within-document term frequencies as well as collection frequencies were therefore computed for the K400A collection to see how the new set of bad discriminators obtained compared with that based on collection frequency only. The two sets were very similar, and retrieval experiments using the new set were therefore thought not worthwhile.

Cross checks on the primary indexing comparisons described above are not available, since the approaches to indexing we have studied here under the heads of term weighting and classification interfere with the initial vocabulary specificity. It may simply be noted that in very early experiments in grouping all the terms in an indexing vocabulary, for the C200I collection, performance for classes for the infrequent vocabulary was superior to that for the full vocabulary, which was in turn superior to that for the frequent vocabulary. This classification comparison thus shows results similar to those for terms alone.

* The former would in any case be deleted by a Q threshold, and the latter could well be.

Alternative performance results for specificity cover tests with quartile deletion for the C200I, I500I and K800I collections (run sets SV3-6), and those removing frequently occurring terms using different thresholds or definitions from the K400A collection (runs SV7-10). The results for the methods except tr taken together show that deleting terms is a mistake, particularly those with medium frequency, and that performance may sometimes benefit only from the removal of very frequent terms. But this factor is one where the treatment of output assumed by the alternative methods is extremely misleading: as the comment above on the recall ceiling suggests, when the tr figures are considered the picture looks rather different. In some cases removing frequent terms reduces the number of non-relevant retrieved without serious loss of relevant documents, but this is not true for the K800I of K400A collection, suggesting that the character of requests is very important here.

## III Indexing Factors

### 1 Statistically Based Indexing

As mentioned in the Introduction, statistically based automatic indexing, i.e. further processing of the primary request and document descriptions, aims to exploit information about term occurrence and/or term cooccurrence patterns. The former naturally leads to term weighting schemes, the latter to classification. In the present section we assume primary document and request term lists, perhaps simply representing all the non-stop words in a document abstract or title, or perhaps the result of eliminating individual document terms to reduce exhaustivity (for example terms occurring only once within abstracts) or of removing some vocabulary terms having specificity properties deemed unhelpful.

Statistical classification exploits more information than weighting and hence in principle allows more ambitious indexing procedures than weighting. The natural order of discussion would therefore be to take weighting first, and then classification. Our experiments with term classification will nevertheless be considered first, partly because historically our research on classification preceded our work on weighting, and partly because automatic classification has not proved profitable and our main project efforts have gone into weighting, which has proved very profitable.

### 2 Term Classification : tight or loose type, substitution or addition use

The project classification experiments followed earlier lines of work (see Sparck Jones 1970, 1971a,b) and were intended to obtain results for more collections than had previously been feasible. It was hoped that sufficient information would be obtained to show whether classifications could generally improve retrieval performance or not, and why it was thus effective or ineffective.

In earlier work tests with the Cranfield data, and specifically the C200I collection, showed that retrieval performance could be improved through the use of a classification, i.e. that exploiting class relations among terms in simple coordination type matching gave better results were obtained for unclassified terms. Detailed investigations showed that controls on the input to classification were required, so that grouping was restricted to infrequent terms (with frequent terms acting as one-member classes) and that classes should be confined to very strongly related items. It was also found that term occurrence information could be exploited to promote recall, or precision, or both, through allowing alternative and additional matches respectively between documents and requests. Indeed it appeared that the precision potential of a classification was as important as, or more important than, the recall capacity it was originally intended to have. At the same time, these tests showed that refinements in the grouping procedure and subtle variations in classification subparameters like the choice of term similarity coefficient or specific class definition, were unimportant. It appeared that if the collection provided a suitable field for classification, relatively simple strategies worked as well as more complex ones (a finding in accordance with those of retrieval research in general).

In subsequent experiments with the Inspec and Keen data, i.e. the I500I

and K800I collections, however, classification did not lead to any performance improvements over simple term matching, though with respect to the different classification variables findings were the same as for the Cranfield material: for example, restricting grouping to infrequent terms gives better performance than grouping all terms.

Attempts were made to account for these collection differences with respect to classification in terms of differences in the intrinsic properties of the collections used (Sparck Jones 1973a). Tests were carried out to see if the Inspec and Keen documents and requests permitted the effective exploitation of a classification in searching: for example if the requests contained few infrequent terms, classificatory information could not be imported to contribute to matching; if the Inspec and Keen data generated a rich and hence potentially useful classification: for instance if the documents contained few terms, only sparse cooccurrence relationships could be established; and indeed if the grouping obtained was statistically significant. Unfortunately it could not be established that there were large differences in any of these respects between the Cranfield data on the one hand and the Inspec and Keen on the other, which could explain the differences in classification performance. At most it appeared that there were some differences of degree, which were reflected in the greater separation between relevant and non-relevant documents indicated by the Cluster Hypothesis Test for the Cranfield collection than for Inspec and Keen.

In the present project it was hoped that by carrying out classification tests on more collections, the conditions for successful classification could be established. The actual procedures for constructing and using classifications were the same as for the previous experiments, since it was thought that the lack of performance improvement should be attributed to the context of classification rather than the classification procedures themselves. It was felt that the tests already conducted had sufficiently established the relative merit of the different techniques. The fact that those which turned out as effective as any were very simple was an advantage for further experiments.

The specific points of comparison in the early classification experiments can thus be summarised, to provide a backgorund for the later tests, as follows:

(1) should all the terms be grouped, or only some? Some.
(2) should a complex or simple similarity coefficient be used? Simple.
(3) should a complex or simple class definition be used? Simple.
(4) should grouping allow weak term relations or only strong? Strong.
(5) should classes be used as sources of substitute or additional matchine terms? Either.
(6) when used for addition should this be done broadly or narrowly? Narrowly.

From these findings it follows that we can adopt a simple automatic classification and class-using retrieval proceudre thus:

(a) select infrequent terms in the vocabulary for grouping, by the 'moiety' threshold referred to in the discussion of vocabulary specificity above. (For thresholds used, see Figure BII.11).
(b) calculate similarity values for the pairs of terms using the Tanimoto (Jaccard) coefficient (see Appendix 1).
(c) form 'stars' as classes, consisting of the terms most strongly connected to each given infrequent term, up to some specified limit of star size. This should be small, so the classes are in fact confined to strongly similar items. For 'stars 2' terms are combined

with their most similar term, or set of equally most similar terms. (Note that the members of a class are mutually related, and further that any given term may appear in a number of classes which may or may not overlap. The logical status of the members of a class is thus the same, and so is that of a particular term when it appears in different classes. We are thus dealing with a genuine, albeit simple, classification and not with a 'semi-classification' in which the information given by each row in the term similarity matrix is treated independently (see Sparck Jones 1974). )

(d) expand requests by adding to them all the terms occurring in any class with each (infrequent) request term, i.e. add in term class-mates.

(e) match the expanded request term lists against the original document term lists.

This procedure is as effective as any of the ones studied, and is relatively easily implemented. A point of particular importance is that the given document descriptions are not modified.

For reference, comparative test results for the C200I collection are reproduced in the Report to illustrate relative performance for the main options listed. Thus runs MC3 and MC4 compare grouping for the full vocabulary and grouping restricted to the infrequent terms, using stars 2 as classes and with the classes used as higher level descriptors replacing the source terms in documents and requests. The restricted result is clearly superior. Runs MC1-2 and MC4-6, all for the restricted vocabulary, compare different class definitions, the simple stars 2, 'strings' 'cliques' and 'clumps'. The clique definition is a stringent one requiring similarities between all class members, and the clump one is a sophisticated one depending on a balance between internal and external similarity connections (see Appendix 1 for precise definitions): a high similarity threshold was applied in both cases; strings are similar to stars but link successive most strongly connected terms up to the natural cutoff of a loop or (rarely) a default length of seven elements. Again using the classes as descriptors, performance for the stars is as good as that for the other definitions. Runs MC2 and MC3 compare stars 2 and stars 6 for the restricted vocabulary, again used as descriptors, showing the former superior. Runs MC4 and MC7 indicate alternative uses of classificatory information given by the restricted stars 2, as descriptors permitting term substitution and as a source of additional terms for expanded requests. Performance is similar. The final comparison, between runs MC7, MC15 and MC16 shows the effects of adding terms (from restricted stars 2) to requests only, documents only, and to both. The first two give similar performance, adding to requests requiring less work; expanding both requests and documents is less discriminating and performance is inferior.

It must be emphasised that though these illustrations are for the hoary C200I collection, many of the comparisons have been made on other collections, with parallel, though absolutely less good, results.

Further experiments during the project were all based on the procedure of (a) - (e) above, using stars 2. It will be noted that the use of the restricted vocabulary for grouping represents a particular response to vocabulary specificity; and the grouping itself affects term specificity, even when terms are simply used to expand requests.

The main objective of the recent experiments was to obtain more information about classification performance with the different collections available. Thus taking the older collections together with the newer ones, we have

tried class-based retrieval on all the test collections except C1400A and U2700CP. (The particular properties of the U2700P collection make straightforward application of the classification procedure impossible). Performance for all the collections is given in run set MC7, with some runs for relevance variants in run set MC8. When taken together for comparison with the primary indexing of run sets MT1 and MT2, differences in the input of mode, source and exhaustivity are disregarded. The comparison shows that the only case where classification performance is unequivocally superior to that of the unclassified terms is the C200I collection, with K800I slightly superior. On the other hand, since only I500I with the high relevance variants was slightly inferior, the overall conclusion must be classes = terms. The lack of improvement for the C1400I collection was the main reason why classifications were not generated for C1400A and U2700OT: it was thought virtually certain that the substantial effort involved would not be worthwhile. More restricted comparisons reflecting constraints on the various input factors shows no more selective differences of performance. The only point worth comment is that where titles specifically are concerned classification performance is virtually identical with that of terms.

As mentioned, in the earlier tests an attempt was made to identify possible explanations for the ineffectiveness of classifications for Keen and Inspec (Sparck Jones 1973). In particular, as it was hypothesised that the relatively exhaustive descriptions for Cranfield favoured classification, while the sparse ones for Keen and Inspec hampered it, performance for randomly reduced descriptions and vocabulary was compared with the original for Cranfield, and that for randomly enlarged descriptions and vocabulary was compared for Keen, using the K400I collection which could be supplemented by information from the corresponding abstracts. (The deletion and addition was confined to infrequent terms). Runs MC13 and MC10, and MC14 and MC12 respectively show that classification remained effective in the first case, and was no more effective in the second.

It was subsequently thought that a possible reason for the ineffectiveness of classification other than for the Cranfield data was that classes were confined to very rare terms by the emphasis of strong similarities. Since requests typically consist of more frequent terms, the class information would never be exploited in searching. Some colour was lent to this suggestion by a detailed comparison between the C200I and C200Ic collections. In the original C200I collection most terms occurring only once were omitted, but were reinstated for C200Ic. Classification is still superior to terms for C200Ic, see runs MC7 and MT1, but is inferior at lower recall to classification when terms occurring once are removed, as run MC9 shows. Run set MC9 also gives performance for the I500I and K800I collections with terms occurring once inhibited from grouping. The K400A and C200A collections were processed in this way too, but with the very full descriptions this would have little effect. The comparisons between the two bases for grouping for I500I and K800I unfortunately show no difference, so the overall conclusions for the comparison between the less and more restricted classification must be less <= more. A related attempt to provide a vocabulary base for grouping by applying the more refined criterion of Salton's Q function, for the K400A collection, was no more successful, as the comparison between runs MC11 and MT1 indicates.

Alternative performance characterisation for classification is confined to that for the simple use of request class-mates derived from stars 2 for a restricted vocabulary. Run sets SC7 give the results for the C200I,

I500I and K800I collections, SC11 for the K400A collection with a slightly different vocabulary restriction. Comparison of the classification run results with those for terms in T1 shows that performance is almost universally the same. This applies to the tr figures too.

These investigative experiments were not carried very far. The fact that classification had no beneficial effects on the C1400I collection was very discouraging, and suggested in particular both that the original results forthe C200I collection were somewhat exceptional, and that many apparent reasons for the lack of improvement with classifications were not the true ones. Further analysis is attempted in Section C. In the meantime the set of experiments considered here may be summarised by classes = terms.

For obvious reasons, no illustrative graphs are supplied for the classification tests.

3 Term Weighting : document or collection or relevance type, ordering or selection use

3.1 Frequency-based Weights

Studies of both indexing vocabularies as such and as the raw material of classification suggest that terms can have different values depending on their statistical distribution. But as noted in the discussion of input factors, permanently deleting 'bad' terms is rather drastic. Further, using the rich and detailed information available for terms only to divide the vocabulary into two classes of term, good and bad, is perhaps not exploiting it for what it is worth. It is natural to suggest that a less crude and more responsive approach is to weight terms by their distributional values in some way or other.

Early SMART Project tests investigated the value of within-document frequency information used in a direct way (Salton 1968a,b). It was concluded that such information was helpful, though its contribution to retrieval performance is often slight. (It at any rate does not degrade performance). In Sparck Jones 1973c experiments with the Keen 400 abstracts showed no material gain from scoring by term weights rather than simple presence in coordination level matching. Mor recent SMART tests (e.g. Salton 1973b) suggest that within-document frequencies may give an extra boost to other forms of weighting, though the contribution is never large.

Within-document frequencies constitute one source of statistically-based weighting. Two other sources may also be exploited, relating terms in individual documents to a document set as a whole. One is document 'description length', used so that the occurrence of a term in two documents is differentiated by the number of other terms or postings in the two. The presumption would be that occurrence in a shorter document was more significant than occurrence in a longer one. The other source is collection 'file length', which allows term differentiation by the number of documents in which each occurs, or by their total postings. The presumption here would be that occurrence in a specific document was more significant if total occurrences were low.

When document descriptions are binary the number of terms in a document is the same as the number of postings, and the number of documents in which a term occurs is the same as its total postings. When document descriptions

are not binary, allowing terms with different within-document frequencies, a distinction can be made between the number of terms and of postings per document, or for a given term, between the number of documents in which it occurs and its total postings over those documents. In general, the greater the posting frequency of a term, the more documents it occurs in, but it may be desirable to distinguish the two explicitly.

For convenience we may define the following quantities:

within-document frequency

$f_{ij}$ = the frequency of term i in document j (abbreviated as f)

posting frequency

$p_i$ = the total frequency of term i over the document collection (abbreviated as p)

collection frequency

$n_i$ = the number of documents containing term i (abbreviated as n)

document length

$d_j$ = the total of within-document frequencies of terms in document j (abbreviated as d)

term length

$t_j$ = the number of terms in document j (abbreviated as t)

We also have

P = the total postings in the collection

N = the total number of documents in the collection

T = the total number of terms in the collection.

If the document descriptions are binary, $f_{ij}$ will be 1 for any present term, $d_j$ will be the number of terms in the document, $t_j$, and $n_i$ will be the same as $p_i$

file length subsumes posting frequency and collection frequency; description length subsumes document length and term length.

These types of frequency information and their relations are considered in Salton 1975b, for example. The theoretical and practical value of the different sources and appropriate modes of exploiting them will be more fully considered in Section C. In the present chapter experiments concerned with on the whole more simple ways of utilising frequency information for weighting are described.

Some simple experiments to establish the relative merits of the three sources of weights described above are reported in Sparck Jones 1973c. Specifically, they compare the use of within-document frequency weights with simple binary descriptions (for convenience these may be labelled unweighted), and weighting by either description length or file length applied to either form of description. The experiments showed that file length is much the most useful base for weighting, the others having little or no effect. Matching scores were computed as the sum of term weights, giving output ordered by notional coordination level.

The specific techniques for deriving weights from the different types of frequency information were straightforward. The within-document frequency of a term was simply adopted as the term within-document frequency weight. For description length weights a simple even scale ranging from 1

for the longest description up to 10 for a document with 1 posting was used (the exact formula is given in Appendix 1). The particular weighting strategy used for file length information was suggested by the earlier studies of term specificity and the treatment of a term vocabulary in grouping, and by the general distribution pattern of a set of index terms. The earlier studies showed that frequently occurring terms should not be removed, but their matching impact should be restricted. The aim of weighting should therefore be to give less frequent terms more favoured status. A systematic and theoretically motivated approach to weighting on this principle then follows naturally from the characteristically Zipfian distribution of index terms, giving the rarest terms the highest weights and the most common the lowest, on a logarithmic scale: i.e. inversely relating frequency and weight. The formula for file length weights is

$$w = - \log \left(\frac{p}{P}\right)$$

or more transparently for binary document descriptions, giving collection frequency weights,

$$w = - \log \left(\frac{n}{N}\right).$$

Successful experiments with this formula (in its second form) were reported in Sparck Jones 1972, indicating a performance improvement ranging from noticeable to material for different collections. This collection frequency weighting has also been studied under the name of "inverse document frequency weights" by Salton, who has obtained similar improvements in performance (see Salton 1973b, 1975b, and 1976).

A related approach to weighting exploiting collection frequency information is represented by the use of Salton's Q function, described above. As indicated there, this may either be based only on collection frequencies, for binary descriptions, or incorporate within-document frequency information as well. The term rankings induced by the two functions are similar in that terms with high collection frequencies have the lowest weights, but differ in that medium frequency terms tend to have higher weights than rare ones for the Q function, while the reverse is true for the logarithmic formula. Salton's use of the Q function for weighting rather than vocabulary reduction represents a development parallel to that outlined above, in seeking to reduce the impact on recall of term elimination and to improve the use made of term discrimination infromation. The SMART experiments reported in Salton 1973b, 1975b show that discrimination weighting based on the Q function leads to an improvement in performance; and that when discrimination weights and collection frequency weights are compared the performance improvements obtained are about the same.

It will be evident that as the sources of weights differ, the corresponding weights may be combined to form a composite term weight. The experiments described in Sparck Jones 1973c combined within-document frequency weights with either description length or file length weights, multiplying the two. Similarly, as Salton typically uses within-document frequency weights, these are combined with collection frequency or discrimination weights. Sparck Jones did not combine description and file length. However, a normalising matching coefficient like cosine correlation, which is ordinarily used by the SMART Project implicitly weights by description length, so all three forms of weighting may be combined. Some tests along these lines are reported below in connection with output factors. There is some evidence in, for example, the results

presented in Salton 1973b and 1975b, that the combination of information is useful, but it appears, as will be discussed in Section C, that the main contribution is made by collection frequency information. This information may indeed be used in ways other than those described so far, and some recent experiments by Salton with these alternatives will be considered in Section C.

Some mechanical implications of the various weighting schemes should be mentioned. Within-document frequency weights are of course associated specifically with the terms in a document, and are normally noted on input. The same applies to within-request weights. File length weights are most naturally calculated at search time, or at any rate at standard intervals for a growing collection. If no reference is made to description length, file length weights need only be calculated for search terms. They are in any case associated with the document terms, and indeed for all the document terms.

The project tests on weighting followed those described above, mainly applying the weighting formulae mentioned to new collections.

The initial comparison is between within-document frequency weighting, represented by run set MW2, and no weighting, represented by the primary indexing MT1. Both document and request weights are used, the collections being the abstract ones, C1400A, C200A and K400A which are comparable in mode, source and exhaustivity. Run sets MW3 and 4 and MT2 and 3 cover relevance variants. The results show that for the coordination type matching in question, where the weights of shared document and request terms are simply multiplied, the weights give no performance improvement, i.e. that no weights = weights.

The results for the crude description length and specifically term length weighting scheme used for the earlier tests are reproduced in run set MW5, for comparison with the primary indexing and also other weighting runs. The tests involved the C200I, I500I, K800I and K400A collections, the first three comparable in mode, but all otherwise differing. The results, compared with the primary indexing, show no improvement, so the conclusion is again no weights = weights. The rather different treatment of description length through normalised matching is considered in Chapter B IV.

The main tests with weighting were with file length weighting, and specifically with collection frequency weights.* The experiments were intended to show that this technique, the only one consistently improving performance in tests prior to the present project, could maintain its value, particularly when used with large collections. Run set MW7 covers all the project collections, including alternative requests, with high relevance variants represented by run set MW8. It was regarded as especially important that the device be tried on the U27000T and U27000P collections. (Adjusted performance for U27000Pb1 is given in Table O). The results unequivocally show that weights > no weights, and indeed usually that weights >> no weights. The set of comparisons as a whole covers variations in mode, source and exhaustivity, and restrictions by one or more input factor simply confirm the general picture.

* in fact in the form - log (n/max n), i.e. with N, the total number of documents in the collection, replaced by the collection frequency of the most frequent term: we have always believed that this makes no material difference, but regret the misleading discrepancy between our theoury and our practice.

We hoped to investigate discrimination weights computed with Salton's Q function, but found it was far from obvious how weights appropriate to simple coordination matching could be derived from the given Q values; and as Salton's own experiments were not particularly profitable, we abandoned the project.

When the forms of weighting are compared, for coordination matching, it is evident that file length weighting, in the form of collection frequency weights, > description, i.e. term length weights.

A few results are available for combinations of the types of weight. Thus for K400A collection, runs MW6 and MW9 reproduce the earlier output for tests combining within-document frequency weights and description or file length respectively. It appears from these runs that devices which are ineffective independently are no more effective in union, and that the only useful contribution is made by collection frequency weights. The three-way combination is represented by normalised matching, below.

These comparisons are somewhat inadequate, though the findings parallel Salton's more extensive tests with weighting combinations. The tests were not taken further mainly because working with within-document frequencies is expensive, since abstract collections are involved, and it was thought unlikely that any striking results would be obtained.

As mentioned before, the dis and cos scoring coefficients used for many of the alternative forms of performance representation incorporate a document length weighting element and so are not strictly comparable with the main results. However, comparisons for the alternative methods between terms with and without types of weight are in themselves permissible where the scoring coefficient for these is the same. The comparisons are in fact confined to collection frequency weights, but performance for these weights has been tested using the alternative methods for all the collections. The results, comparing run sets ST1 and SW7 show a general tendency for the weighted terms to perform better than the unweighted ones. The same tendency appears in the results for the sum scoring coefficient, which does not normalise.

In general, the overall conclusion from the project experiments is that the only effective and cheap form of weighting is that exploiting term collection frequencies, for which material performance improvements have been obtained. Thus the tests may be summarised by collection frequency weights >(even >>) no weights.

The use of term length weights is illustrated in Figure BIII.1, and of collection frequency weights in Figure BIII.2.

## 3.2 Relevance Weights

The weighting schemes using term frequency just described are based primarily on the occurrences of terms in documents. The documents in a collection all have the same status as sources of weighting information, though the actual information derived from them will vary in value.

It has been suggested that statistically-based weighting schemes can be extended, and made more powerful, if additional information is supplied which reflects differences in the status of documents, and specifically their differences in relevance, i.e. whether they are relevant or not. A

term in a request may then be evaluated by its past success in retrieving relevant documents. More precisely its likely value in retrieving relevant documents in the future may be predicted from its relative occurrences, for past searches, in relevant documents and in any documents, and a relevance weight be assigned accordingly.

Robertson 1974 suggested that collection frequency based weighting could be vamped up in this way, and following earlier, tentative proposals by Barkla,and experiments reported in Miller 1971b, put forward an appropriate weighting formula. A similar idea was studied by Barker 1972b, 1974 as an aid to manual SDI profile amendment, with further work by Robson 1975, 1976; and recently experiments with relevance weighting have been carried out by the SMART project (Salton 1976, Yu 1976). The requirements for relevance weighting are clearly greater than those for the schemes discussed above: the latter depend only on information which can be supplied for the collection about to be searched. Relevance information clearly cannot be supplied directly for a collection before it is searched, but under broad assumptions of consistency in material and needs, past output, if statistically adequate, may be exploited for future searches. The approach would therefore be appropriate for an SDI system with some user feedback. More importantly, it would fit very well into iterative, onl-line searching, where relevance data associated with one search cycle could be exploited in the next.

In general, it may be expected that as past experience accumulates, prediction for the future is more accurate. A natural limit is indeed reached if such a formula as that suggested by Robertson is applied retrospectively; and Sparck Jones 1975 suggested that Robertson's formula, so applied, could be used to provide a performance yardstick for test collections where relevance information is already available. In this case weights are computed from the known occurrences of request terms in both documents in general and relevant documents in particular, and then used in a re-search of the collection to determine how effectively the given relevant documents could have been retrieved with the given requests. Performance is optimal for the weighting formula; and it may be treated, more broadly, as indicating a level of performance better founded than that represented by the conjunction of 100% recall and 100% precision, to which searches based on simpler methods, or less complete information, might aspire: this use was discussed in Chapter II.3 of Section A.

This line of work has been carried much further in the present project, and extensive experiments have been carried out. In Robertson 1976 a series of relevance weighting formulae are derived from first principles. Thus two different initial assumptions may be made about the relative distribution of request terms in relevant documents and in documents in general, and two different ordering principles may be invoked to organise search output according to the occurrence of query terms in documents. Thus in the first case we may adopt either

Independence Assumption I1: the distribution of terms in relevant documents is independent and their distribution in all documents is independent;
or
Independence Assumption I2: the distribution of terms in relevant documents is independent and their distribution in non-relevant documents is independent.

In the second case we may apply either:

Ordering Principle O1: the probable relevance of documents is based only on the presence of search terms in the documents;
or
Ordering Principle O2: the probable relevance of documents is based both on the presence of search terms in the documents and their absence from the documents.

Either Assumption may be combined with either Principle; and each of the four possible combinations leads to a specific relevance weighting function F1 - F4, as follows:

| | | Independence Assumptions | |
|---|---|---|---|
| | | I1 | I2 |
| Ordering Principles | O1 | F1 | F2 |
| | O2 | F3 | F4 |

To define the function we take the definitions of section 3.1 to refer specificially to query terms i, and add to them:

relevance frequency

$r_i$ = the frequency of term i in a query over the set of relevant documents for the query (abbreviated as r);

R = the total number of relevant documents for the query.

We now define the weight of term i in a particular query, according to the different choices of Assumption and Principle as

$$w = \log \frac{(\frac{r}{R})}{(\frac{n}{N})} \qquad \text{F1}$$

or

$$w = \log \frac{(\frac{r}{R})}{(\frac{n-r}{N-R})} \qquad \text{F2}$$

or

$$w = \log \frac{(\frac{r}{R-r})}{(\frac{n}{N-n})} \qquad \text{F3}$$

or

$$w = \log \frac{(\frac{r}{R-r})}{(\frac{n-r}{N-n-R+r})} \qquad \text{F4}$$

The various quantities occurring in these formulae refer to the simple contingency table for term i:

| | | Document Relevance + | Document Relevance - | |
|---|---|---|---|---|
| Document indexing | + | r | n-r | n |
| | - | R-r | N-n-R+r | N-n |
| | | R | N-R | N |

While the different pairing of Assumptions and Principles can be made, as indicated, it is argued in Robertson 1976 that the combination of I2 and O2, yielding F4, is the correct one. F1 is the formula originally suggested by Robertson 1974 and used for the tests reported in Sparck Jones 1975; it has an obvious relationship with the collection frequency weighting formula discussed in 3.1, which may for convenience be referred to here as FO. All four relevance weighting formulae were compared in Robertson 1976, the experimental results supporting the theoretical arguments in favour of F4. More specifically the tests showed that the alternative choices of Independence Assumption made little difference, but that the choice of Ordering Principle is important, with F3 and F4 performing better than F1 and F2. Tests since have chiefly involved F1 and F4, the most opposed formulae.

It must be emphasised that these formulae weight a query term in relation to that query only, i.e. that the same term may have different weights in different queries. The relevance weighting scheme studied by the SMART Project, which has also been supported by formal arguments (Yu 1976), assigns a relevance weight to a query term on the basic of its average effectiveness in different queries.

The experiments reported in Robertson 1976 were designed on the one hand to illustrate more fully the retrospective use of the relevance weighting formulae, to provide performance yardsticks; and on the other to study their more important predictive use. The retrospective tests were on the C1400I and K800I collections: they showed a strikingly higher level of performance for F1 and F2, compared with FO (which may be regarded as indifferently retrospective and predictive), and a much higher level still for F3 and F4. The predictive tests were carried out using the C1400I collection divided into equal even and odd-numbered document sets (the C1400Ie and o collections); weights were computed using information derived from the even set and applied in searches of the odd one. These showed a noticeable difference between F1 and F2 on the one hand and FO on the other, and a further material improvement for F3 and F4 over F1 and F2. Comparable retrospective performance for the odd documents was of course superior to the predictive.

Analogous experiments by the SMART Project reported in Salton 1976 showed rather little gain for relevance weighting; but this may be explained by the particular way the weights were applied, since they reflected averaging, as indicated above, and were only computed for a small proportion of the query terms.

In the tests with F1 - F4, the formulae were treated slightly differently for retrospective and predictive application. Specifically, in the second case allowance has to be made for uncertainty, which is achieved by adding 0.5 to all the elements of the formulae. In the retrospective application limiting cases represented by zero values for different components of the formulae were dealt with by extreme measures, forcing documents for query terms with the special properties to the top or bottom of the output ranking. (Details of the limiting cases and their treatment are given in Robertson 1976 and reproduced in Appendix 1). Dealing with these cases is computationally tedious, and expensive, particularly for F3 and F4 which take account of term absence as well as presence. There is some justification for the view that even in the retrospective case, statistical uncertainty exists, and the predictive version of the formula may be adopted for retrospective application. We thus distinguish predictive applications of

the weighting formulae from the retrospective application of their predictive form. The predictive versions perform less well than the original in retrospective tests, particularly for the smaller collections, though the size difference may be accidental. This is not surprising since they treat limiting cases less favourably. The comparative performance for the original and predictive versions of F4, for the C1400I and Io, C200I, K800I and T, and U27000T, Tl and U27000Pbl collections, is given in run sets MR9 and MR13. The original versions of the formulae are labelled Fla and F4a, since arbitrary large weight values were assigned to terms with special properties, to achieve correct ranking consequences.

Project tests with the weighting formulae were intended on the one hand to provide further information about the test collections through the retrospective 'yardstick' searches, with further comparisons between Fl and F4; and on the other to carry out some serious predictive experiments with large collections, and in particular with the UKCIS material.

The results for retrospective searches with the original versions of formulae Fl - F4 i.e. Fla - F4a are given in run sets MRlO-13, for the C1400I and C1400Io, C200I, K800I and K400I collections: these share the smae indexing mode but differ in source and exhaustivity. The runs all show Fla much superior to FO (collection frequency weighting, itself superior to simple terms), with F2a performing much the same as Fla; and F4a much superior to Fla, again with F3a similar to F4a. An extended range of comparisons between Fla and F4a, for nearly all the collections, and disregarding input factor variation, is given in run sets MRlO and MR13. Except for the U27000T collection, for which they are not available, the runs are all for manually indexed requests. The overall picture is much the same as for the runs just described, with Fla universally much superior to FO, and F4a to Fla, though in a few cases, C1400A and T, F4a is not superior to Fla. The overall conclusion must therefore be that FO << Fla, and Fla < F4a. Possible reasons for the variation are considered in Section C.

The predictive experiments previously reported for the C1400Io collection are reproduced in runs MR3-7. For tests with the UKCIS material, the documents were divided into the first 'half' and the last 'half' (the latter somewhat larger, with 15748 documents: see Figure AII.7). The division constitutes a stringent test of the predictive weighting, since it corresponds to gross subject groupings in Chemical Abstracts Condensates. The initial requests for the U27000T and U27000Pb collections are of course very different in character, and the set for the latter is smaller as it consists only of the 75 profiles with an original strict Boolean specification. The results run sets (with some adjusted versions in Table O) show the same pattern of performance as the tests with the Cranfield data, though performance for U27000Pb is absolutely much better than for U27000T. Thus terms are much inferior to FO, which is in turn inferior to Fl for C1400Io and U27000Pbl, though not for U27000Tl, while Fl is in turn inferior or much inferior to F4. The predictive performance may be compared with the corresponding retrospective results of run sets MR9-13. The value of predictive weighting itself may be summarised by FO < Fl, and Fl < F4.

The different collections involved in these tests vary with respect to input factors. Relative performance for relevance weighting for the different collections derived from the same material generally parallels that of the primary indexing: i.e. it does not appear that the response of

relevance weighting to input factors is different from that of the underlying simple term matching.

The combination of term weights with Boolean search specifications, which was studied for the U27000Pb1 collection, is considered below.

Performance for the retrospective application of relevance weighting has already been shown in connection with the establishment of yardsticks in Figure AII.18 for the C1400I and K800I collections; further illustrations, with a comparison between the retrospective application of the predictive formula and that of the formula with special cases, appear in Figure BIII.3. Predictive performance for the C1400Io, U27000T1 and U27000Pb1 collections is shown in Figure BIII.4. Some adjusted results for the last collection appear in Figure BIII.5.

Other indexing devices using relevance information

The weighting techniques just described all relate the behaviour of query terms in relevant documents to their behaviour at large. A much cruder scheme is simply to weight request terms by their basic relevance frequency r. For reference we may label this 'formula' $F^{1}_{2}$. A few runs have been carried out with it, mainly to test the value of the normalisation introduced in F1 - F4. Its retrospective application is tested for C200I and K400I in runs MR2, and its predictive application for C1400Io in run MR1. The retrospective use shows no improvement over term performance for C200I, but an improvement comparable with F0, thoughmuch inferior to the other formulae, for K400I. A first predictive test with the formula as given had no effect, compared with term matching for C1400Io; although it is surprising that performance was not degraded as many query terms had weights of 0 since they did not occur in any relevant documents in the even set from which the weights were derived. It was, however, thought that it would be more appropriate in the predictive case to allow all terms to have a weight of at least 1, but the test results for this, those shown in run MR1, show no gain.

The general superiority of the formulae described earlier is borne out by some other experiments with relevance data. These are connected with the feedback studies of the SMART Project (see, for example Salton 1971). The ideas involved deserve further, larger scale investigation than they have received hitherto. The tests are mentioned here merely to illustrate other possibilities, and also to show that naive uses of relevance information are not necessarily productive.

These schemes differ from those considered hitherto in adding terms to the request: in relation to the term composition of requests, the weighting devices can only simulate term deletion, and are essentially precision devices: indeed it is of interest that the original versions of the formula when used retrospectively may actually reduce recall slightly. An obvious approach to stimulating recall, using relevance information, is to add terms taken from known relevant documents to the request. One naive way of doing it is to add in all the terms from any known documents, and since there may be many of these, to weight them with $F^{1}_{2}$ to introduce some value distinctions. As the resulting requests tend to be very exhaustive the retrospective application is meaningless: the result for a predictive application (with weights minimally 1) is shown for the C1400Io collection in run MR3; the approach is unexpectedly ineffective, perhaps because with this data, the requests are far too exhaustive.

An alternative is to use a single relevant document as a request, either choosing one at random (say the numerically first), or the highest matching one known. Experiments with the first alternative are illustrated, retrospectively applied, for the C200I, I500I, K800I and K400I collections in run set MT5, and for the second for C200I and K400I in run set MT7. Predictive tests with both for C1400Io are given in runs MT4 and MT6. Rather oddly, the retrospective cases show less good performance than the original requests, especially for the 'random' approach; and the same performance as the regular requests in the predictive case.

The simple-minded approaches to using relevance information just described are clearly inadequate. In particular it would seem that allowance should be made their effects on exhaustivity. The tests have been mentioned primarily to indicate some of the possibilities other than those discussed earlier. Whether performance, especially in relation to recall, can be improved by more careful methods of adding relevant terms or by combinations of addition and the sophisticated weighting rperesented by F1 - F4, are questions deserving further study.

The alternative treatments of relevance weights mainly involve sum scoring as the rationale for combining weighting by formulae F1-4 with cosine correlation is not clear (though it may be noted that when combined, however improperly, better performance results). The retrospective use of relevance weights for the C200I and K800I collections shows the relevance weights materially better than unweighted terms, for all the alternative forms of representation. The predictive results with sum for the C1400Io collection show the relevance weights noticeably better than terms. The simple formula $F^{\frac{1}{2}}$ can be used with cosine as well as sum as can the related methods of combining $F^{\frac{1}{2}}$ and relevant document terms, and of just using relevant documents as requests. The two coefficients give somewhat different results: there is generally little improvement over terms with sum, and performance for the enlarged weighted requests is inferior. But this is misleading as sum was mainly intended for use with the more sophisticated relevance approaches which involve their own normalisation. It is noticeable that with cosine the enlarged requests perform better than terms, and the enlarged weighted ones much better.

Predictive use of the enlarged weighted queries, and of the alternative use of relevant documents as queries, is illustrated in BIII.6, with performance for the simple weighting scheme $F^{\frac{1}{2}}$ for comparison.

# IV Output Factors

## 1 Searching

### 1.1 Scanning, Matching, Scoring

As discussed in Chapter III of Section A, a number of system variables have been grouped together under the heading of searching and assigned to Output Factors. In some cases, for example, in relation to request characterisation, the separation between indexing and searching is not particularly obvious, and the allocation of topics to one or the other is somewhat arbitrary: thus in this case the properties of individual query terms are considered under indexing, while those of request specifications as wholes are assigned to searching.

In Section A we listed three main search variables: scanning, the way the document set of documents is inspected for a request; matching, the way individual documents are viewed in relation to a request; and scoring, the way documents are valued as matching. We also noted that while specific search procedures naturally give rise to particular forms of output, unordered or partially or fully ordered, output presentation should be treated as independent of searching since one form may be transformed into another for convenience or comparative purposes.

Some sort of overall framework is required to characterise and relate the many different procedures which may be adopted for scanning, matching and scoring. In the project we have studied several rather different approaches, and linking these with those investigated by other projects makes a fairly comprehensive descriptive framework necessary. The following seems reasonably satisfactory.

We assume document and request descriptions in the form of simple terms. This is for convenience: the schema is applicable to any descriptive entities, whether these are higher level descriptors or complex term structures involving linguistic rather than logical relations between terms. We may then say that a document having something in common with a request, i.e. sharing at least one term, may be regarded as a potential candidate for selection.* Whether such a document is finally selected depends on the specific searching procedure used: possible procedures may be categorised in terms of choices under the three headings of scanning strategy, matching condition, and scoring criterion, as follows.

1. Is any document which is a potential candidate for selection a nominal candidate, or are only some documents nominally candidates; i.e. is the searching of the document collection as a whole exhaustive, or is it partial, as it may be in cluster-based searching where there is no presumption that any document sharing a term with a request will be inspected?+ Note that this is a logical remark: searching an inverted file is exhaustive in the sense in question though not every document is inspected. Again in practice searching may be confined to a sample of a large set, but this does not mean that the formal search procedure is necessarily restricted. We may summarise the options as

scanning strategy: A all documents
B some documents.

* This definition does not allow for the 'anti-matches' of relevance weighting, i.e. looking for the absence of a specific request term; but it could clearly be extended to cover such cases.

+ indeed in cluster-based searching nominal candidates may not possess a request term: the schema would again have to be extended here to cover related terms.

2. Does any subset of the request terms define a match, or only some terms; i.e. is any nominal candidate for selection regarded as matching the search specification, or only some? Simple term coordination searching for a match on one or more of n terms illustrates the former, Boolean or quorum searching requiring a match on more than one term, the latter. Matching documents may be called actual candidates. It must be emphasised that this distinction is considered here primarily from the point of view of documents, that is as having to do with document status in relation to requests rather than with request surface form in itself. These two options may be summarised as

matching condition: A any terms
B some terms.

3. Are all the matching documents treated pari passu, as equally worthy of final selection, as in ordinary Boolean searching (without screening), or are some more worthy than others; and if the latter is the difference between documents due to

(i) multiple matching, e.g. of different numbers of coordinate terms, or of different numbers of group members with a Boolean specification;
(ii) other features of the document not determining the matching, e.g. weights reflecting term frequencies within documents, or document length;
(iii) other features of the request not determining the matching, e.g. weights for request terms?

From (i), (ii) or (iii) a particular score can be derived for the matching documents to generate an ordering. The three are of course not exclusive and may be variously combined in different procedures with appropriate scoring algorithms. As under 2, the emphasis is on documents satisfying a request, rather than on the form of the request itself.

The basic options here may thus be summarised as:

scoring criterion: A equal terms
B unequal terms
due i) number and/or
ii) document values and or/
iii) request values

The final selection of documents to be deemed retrieved in principle follows logically or relatively naturally from the choices adopted under 1-3. Thus the final selection of successful candidates can be made on logical grounds when all the documents matching a Boolean specification are taken, and in a natural way with weighting schemes when all documents having a positive weighting score are selected. Previously specified weight thresholds, as used for some of the UKCIS profiles, have the same status. In all these cases whether a document is finally selected follows solely from its individual relation to the request: they may be contrasted with arbitrary bases for selection like specified numbers of documents or recall levels, from which no prediction can be made as to whether an individual document will finally be selected. Any document ordering introduced by 3 i),ii) or iii), whether used for system selection or not, may also be exploited by the individual user; but any ad hoc cutoff he applies is not in question here. It will be evident that the relationship between the sets of documents involved in the (logically) successive steps may differ: for some search procedures the sets of potential candidates for selection, nominal candidates, actual candidates, and successful candidates will be identical; for others each will be a subset of the previous one; or more complex relationships may hold.

The main headings and choices, i.e. primary variables and value sets, listed define a range of possible types of search procedures which is summarised and illustrated in Figure BIV.1. Essentially the alternatives are based on different views of what is important in establishing the retrieval relation between a document and a request. Thus the choice of scanning strategy for the collection under 1 (when not purely economically motivated) reflects on the one hand the idea that documents should be individually considered in relation to requests, and on the other the notion that they are related as groups, especially of probably co-relevant documents, to requests. The matching conditions of heading 2 deal with the content of a match, either considering request terms occurring in a document individually or taking them together in some way. The scoring criteria of heading 3 determine document merit, absolutely or relatively, responding to different attitudes to document relevance. Particular combinations of choices lead to particular types of procedure. For example, option A, full search, under 1, together with 2B, matching on some terms, and 3A, regarding terms as equals, gives us Boolean searching.

As the figure brings out, the assumptions on which particular types of procedure are based, and the natural final output of a search, differ so much in kind that valid performance comparisons are limited. For example, cluster-based searching is intended not to be exhaustive, so comparing it with exhaustive searching involves a different collection size, which means that relative performance needs cautious interpretation: is 25% recall on a search of half the collection better than 40% with the whole one, for instance? Similarly, to compare procedures which allow an ordered output with those which do not naturally generate one, some adjustment may be required which may be artificial or effectively suppresses the presumed advantages of the procedures in question.

1.1.1 Scoring coefficients

Scoring, particularly when variable scores are allowed under 3B, involves a formal storing coefficient, which may be relatively obvious, as in ordinary term coordination, or more complex, as in Salton's use of cosine correlation. Different coefficients may be used to derive the actual scores from given information: for example, Salton's cosine correlation or van Rijsbergen's normalised symmetric difference (van Rijsbergen 1975a). The choice of scoring coefficient is thus a subsidiary system variable, with more scope in some cases than others. For reference the cosine and difference formulae are given in Appendix 1.

Sometimes a particular scoring coefficient follows very naturally from the underlying logic of the search procedure. This is clearly seen with simple term coordination, where any scoring method other than a count of matching terms seems artificial. Again, for Boolean searching, the scoring coefficient is effectively determined by the search formulation. But sometimes, especially where weighting information is concerned, different coefficients have been advocated: thus Salton originally compared cosine correlation and logical overlap (Salton 1968a,b) while van Rijsbergen has advocated a slightly different normalising coefficient. The choice of coefficients is only genuine when the same information can be digested, and the measures are comparable in objective: thus cosine correlation and symmetric difference are both applicable to binary descriptions, and both normalise. Scoring coefficients apparently differing merely technically may in fact embody different assumptions associated with different choices under heading 3: this would be the difference, for example, between the use of Jaccard's coefficient for matching and simple coordination.

On the other hand, particular types of information may exclude free choices of coefficient: the relevance weighting schemes discussed in Chapter III imply notional coordination level matching and cannot obviously be combined with e.g. cosine correlation.

1.1.2 Output formats

As mentioned in Chapter A III.2, it is important to distinguish the search output for finally selected documents as it is generated by the system from the output as presented to the user. The scoring criteria, 3A and B, generate respectively unordered and ordered output, but the former may be ordered, e.g. randomly, or by accession number, for offering to the user, say in on-line searching; while the latter may be processed by the application of a cutoff, for instance to provide the user with an acceptably small bloc of documents. We said in Chapter A II.2 that the initial material generated by the search system in itself has a specific output form, natural to the search procedure; while whatever is presented to the user, which may or may not embody modifications to the original output, is an output type. Different procedures may generate output of the same form, for example, if the same choices are made under headings 2 and 3 but different ones under 1; and equally output of a given form may be modified to give output of different types.

It is unlikely in practice that an operational system would involve marked differences in form and type: minor modification, say to take only part of a large ordered output, is more likely. But the distinction between form and type is useful for comparative purposes and evlaution since output in a given form may be recast to relate it to other output. Comparisons will thus be between types of output which may be the same as, or distinct from, that originally generated.

The main division is, as indicated, between unordered and ordered output. We also distinguished, in Section A, partially ordered from fully ordered output, and noted that it has sometimes been found convenient to treat an entire collection as retrieved, to give a complete as well as full ranking (the artificiality of this proceeding is well indicated by the fact that it makes nonsense of the search procedure analysis just presented). In general, output transformations should be treated as concerned with ranks rather than matching values. The most useful transformations for comparative purposes are from ordered to unordered output, and from unordered and partially ordered to fully ordered. The former is achieved quite straightforwardly, and as described earlier, the latter by random distribution of equally-ranked documents over the appropriate number of rank positions.

Specific procedures

Figure BIV.1 summarised the main descriptive framework suggested for searching, without going into details under 3B. The options here clearly cover a large number of different approaches, or rather classes of approach representing subvariables defining subtypes of the main types. Figure BIV.2 details the combinations of choices possible here, and lists approaches studied by the project falling under an initial choice of exhaustive rather than partial collection scanning. (Since partial scanning has not been investigated directly in the project research, remarks on it are confined to cross project comparisons considered in Section C). Where more than one entry appears at the intersection of a row and column in Figure BIV.2, this means that the entries concerned satisfy the same formal output characterisation, but differ in content. As noted in the table

itself, the entries themselves may be a shorthand for a group of similar approaches: thus "description length" refers both to term length weighting and document length weighting. The members of a group may also be further differentiated, as indicated, by the application of different scoring coefficients. Though we have not tested them, there are candidates for some of the gaps in the table, though not obviously for others. We have not attempted an exhaustive treatment of this whole topic to provide a full analysis of the elements of our various procedures and of related ones used by e.g. the SMART project here, as more detail is involved than is justified by our actual range of tests.

Figure BIV.2 thus brings out the fact that though we may refer under output factors to devices, like term weighting, we have already considered under indexing, we are now looking at them from a quite different point of view, and are distinguishing and relating them in different ways. In the accounts of our experiments so far, we compared values of particular input or indexing variables in the context of a single search procedure, or at any rate type of procedure defined by the main output variables and their values of Figure BIV.1. The tests were all based on procedures of type 2, reperesenting scanning strategy 1A, matching condition 2A and scoring criterion 3B. As indicated at the end of Paragraph 1.1, Chapter B II, cross checks representing explicit alternative major output factor choices were not generally possible though changes in subtype might be involved in the application of different methods of performance evaluation: for example the use of SMART cosine correlation representing subtype d, i.e. options a + b for recall cutoff might be compared with simple coordination representing subtype a.

Overall we may summarise the characterisation of output factors given in Figures BIV.1 and 2 as follows. We distinguish three main variables, 1 - 3, scanning strategy, matching condition and scoring criterion. Each variable has two values, A or B. The combinations of variable values thus define eight type of search procedure, 1-8. Those types involving option 3B are divided into seven subtypes, a-g, according to their bases for document ordering, where each subtype may represent a group of procedures differing in detailed content rather than form, and possibly, further, by specific scoring coefficient. In the following sections such comparisons across types as are possible with are data are presented. The alternative treatments of performance are, however, of limited use here as they may necessarily cut across the distinctions to be made.

1.2 Scanning Strategies : all documents or some documents

As mentioned, the project has not been concerned with partial scanning, and specifically with cluster-based procedures of the kind studied by the SMART Project and van Rijsbergen, involving motivated document groups rather than arbitrary sampling for economic reasons. Of the four types of search embodying partial scanning, the last two, 7 and 8 illustrated in Figure BIV.1 by Boolean and van Rijsbergen clusters and Boolean and SMART clusters repsectively, have not actually been tested, but may well be in the near future. The difference between van Rijsbergen's and the SMART Project's use of clusters lies in the fact that the former produces unordered output covering all the documents in selected clusters, while the latter ranks documents for retrieved clusters. An attempt to relate the cluster experiments of types 5 and 6 carried out by van Rijsbergen and the SMART Project with those of the present project will be made in Section C.

It may be noted here that, with respect to alternative treatments of performance, cumulative effectiveness was originally advocated as a device for comparing partial cluster based search performance with full search. So where results with it show performance improvements for full search, for example, through the use of weights, these present a challenge to cluster-based searching in offering a higher standard to be matched, and raise the question of whether such techniques can be combined effectively with the use of clusters.

1.3 Matching Conditions

Procedure type 1 represents an output option which has not been systematically studied, but this is hardly surprising as it is rather undiscriminating: it is represented in practice by simple searches on single query terms. The main project experiments have been with type 2, combining matching condition 2A and scoring strategy 3B, i.e. with coordination level searching. Tests with types 3 and 4 have been very limited, and have been confined to the UKCIS data for which appropriate requests, i.e. the profiles, are available.

In principle, comparisons covering the output options for full searching would be as follows: between types 1 and 2 and between 3 and 4, to illustrate the effects of different scoring criteria for the same matching conditions. But it is obvious that a strict interpretation of such comparisons would be futile since it would involve abandoning the output form differences they are designed to exhibit, i.e. 2 would be reduced to 1 and 4 to 3. Only more informal comparisons seem appropriate. The contrast between 1 and 2 applies to all our test collections. If performance for the retrieved set of documents as a whole is compared with that for the ordered set, the familiar recall - precision relationship is exhibited. At higher positions in the ordering representing smaller sets, recall decreases while precision increases. The only noteworthy point is the fact that precision sometimes does not increase very much, or only at the lowest recall. This applies especially to the larger collections. The situation is much the same for any specific choices of option under 3B for 2: for example the choice of subtype a represented by the primary indexing, or of e represented by collection frequency weights, in run sets MT1 and MW7 respectively. The comparison between 3 and 4, restricted to the UKCIS collection U27000P, shows a less familiar situation. Runs 01, 02, 03 and 04 in Other Table on U27000Pb1 show that precision does not generally increase as recall decreases. Thus different subtypes under 4 show different performance curves, and no common trend in either the raw or the adjusted cases 010 or 011. Even where precision does increase, the increase is not striking.

The other pairs of comparisons are perhaps more meaningful, but are again confined to the UKCIS material. In this case there are no differences of output form, but there are variations in the size of the document set retrieved. Contrasting 1 and 3, we find that the inverse relationship is again exhibited, but somewhat skewed. The raw results (013 and 011) are certainly vititated by the unknown relevant documents problem, and the adjusted ones seem preferable. But even here, the striking gain in precision for little loss of recall exhibited by 3 is somewhat surprising, as runs 014 and 09 of the Other Table show. The same applies to the comparison between 2 and 4, for the various specific choices of subtype

under 3B. In each case the improvement in precision for comparable recall exhibited by 4 is very marked, and is maintained to a non-trivial recall ceiling, even for the adjusted runs. The various comparisons are between runs MT1, MW7 and MR7 and O2, O3 and O4, with illustrative adjustment in runs O8 and O10.

For these comparisons it is only appropriate to consider the standard performance representations based on averaging across matching values: the ranking methods we have used all involve complete ranking, which is particularly misleading where investigations of this output factor is concerned.

1.4 Scoring Criteria

Comparisons between the major choices of A and B, generating unordered and ordered output, were involved in the comparisons made in the last section. Those to be considered now are between the alternative subtypes under 3B, chiefly in relation to procedures of type 2.

Unfortunately comparisons even here are affected by differences of output form. As mentioned in Section A, scoring coefficients like cosine correlation generate a fully ordered output, and one for which the only natural base for averaging is across ranks. The output is also completely ranked , so comparisons using it have to be treated with caution here. The main use of this coefficient has been for recall cutoff performance representation, which will be considered separately below.

As appears in Figure BIV.2, some of the seven possibilities are not represented in our experiments. Our comparisons are therefore between a, multiple matching only; d, multiple matching combined with document features; e, multiple matching with request features; and g, the combination of all three. The individual possibilities listed for these subtypes in the table were initially considered, from the point of view of the sources for weighting, in the discussion of indexing factors in Chapter III. As noted there, for the regular output comparisons by matching value, it appears in general that the exploitation of document features like within-document term frequencies or/anddescription length (as we crudely treated it) is of no particular value: i.e. options of subtype d do not ordinarily perform better than a, as run sets MW2, MW5, MW6 and MT1 show. We found, on the other hand, that subtype e in the form of collection frequency weights was superior and in the form of (predictive) relevance weights was much superior, to a. In our experiments option g is represented only by the combination of within-document frequency and posting frequency information, (MW9) which was no better than the latter alone.

Subtype comparisons for procedures of type 4 have been limited to the UKCIS profile data: runs O2, O3 and O4 of Other Table compare option a, simple term coordination ordering of output selected by a Boolean profile specification with e, in the forms of collection frequency and relevance weights. The relationships between a and e are the same as for type 2, but as indicated earlier only relevance weighting gives any overall improvement, and a very small one, over the initial Boolean output.

For scoring criterion comparisons, unlike the matching conditions ones, our alternative performance evaluation technique may be also considered. In general, the results for document cutoff averaging across ranks in the fully ordered output confirm those just presented, though the range of

comparisons (for incidential practical reasons) is not quite the same. Run set SdrT1 for sum and cos compares subtype a, simple coordinate output forced to a full ranking with type d representing description length weighting in the more sophisticated form associated with cosine correlation. The description length weighting here again does not seem to contribute much: it is only of use for very long queries, as in runs SdrT4 and 6. Comparisons with subtype e, representing collection frequency, and relevance, weights combined with description length information in cosine scoring again show performance gains through the use of query features. Unfortunately, subtype e is represented only by the combination of cosine correlation description length and collection frequency or rather crude relevance information: as mentioned earlier, it is not clear that this scoring coefficient can properly be combined with relevance weights of types F1-4. Gains with the use of collection frequency information parallel those for the main coordination matching runs, though the crude relevance weighting is as ineffective as before. It is of interest on the other hand that when queries are enlarged and also supplied with simple relevance weights cosine matching is superior again to coordination (run SdrR2).

No runs using this method of performance evaluation have been done for procedure type 4.

Recall cutoff evaluation, for type 2, makes the same comparisons as those just considered, and supports the same conclusions, as do the other methods of performance representation covered by the secondary tables. Performance generally parallels that of the main tests, the only exception being the improvements obtained with cosine description length for the document based queries of runs ST4 - 6 and SR3. The description length experiments of main runs MW5 and MW6 did not cover such a case.

Taking all the results presented together, the overall conclusion to be drawn is that performance differences are due more to weight content than to the output factors we have been studying. It is true that multiple matching alone tends to perform less well; and it also appears that document features are less useful than request ones. But wide variations in performance are associated with particular choices of request feature.

1.4.1 Coefficients

We have interpreted comparability with respect to scoring coefficients rather strictly, in requiring them to have the same inputs. Our only, fairly restricted, tests have therefore been confined to contrasting SMART Project cosine correlation with van Rijsbergen's normalised symmetric difference. Results for recall cutoff evaluation, with which the advocates of these coefficients have been mainly concerned, are given in Secondary Tables Src. They show that where there are differences, cosine correlation is superior to symmetric differences, but the differences are not large.

It will be evident from the foregoing that for output factors, a separate treatment of alternative methods of performance representation is not in order.

The output options discussed may be illustrated by the graphs of Figure BIV.3 showing coordination searching, Boolean, and a combination of the two, for the U27000Pb1 collection, with adjustments in Figure BIV.4; and by Figure BIV.5 one for document ranking based on various scoring criteria, for the C1400Io collection.