DOCUMENT RESUME

ED 032 216 SE 007 469

By-Jackson, David M.

The Construction of Retrieval Environments and Pseudo-Classifications Based on External Relevance.

Ohio State Univ., Columbus. Computer and Information Science Research Center.
Spons Agency-National Science Foundation, Washington, D.C.
Pub Date Apr 69
Note-81p.
EDRS Price MF-$0.50 HC-$4.15
Descriptors-*Classification, Computer Science, Computer Storage Devices, *Information Retrieval, *Information Science, Models
Identifiers-Computer and Information Science Research Center, National Science Foundation, Office of Scientific Information Service

The idea of pseudo-classification based on external relevance was introduced and compared with the usual classifications using associative techniques. A general model for an information retrieval system using term classification was described. A set of operators for adjusting pseudo-classifications and preventing their deterioration was derived for a particular match function conforming with this model. Pseudo-classifications were used for prediction of relevant documents and for the evaluation of retrieval systems with respect to their theoretical optimum. The concept of improvability of a retrieval model with respect to its constituent submodels was introduced and elaborated upon. (RR)

CISRC

F-NSF

TECHNICAL REPORT 69-3

THE CONSTRUCTION OF RETRIEVAL ENVIRONMENTS AND PSEUDO-CLASSIFICATIONS BASED ON EXTERNAL RELEVANCE

by

David M. Jackson
U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION

APRIL, 1969

COMPUTER and INFORMATION SCIENCE RESEARCH CENTER

The Ohio State University
Columbus, Ohio 43210

THE CONSTRUCTION OF RETRIEVAL ENVIRONMENTS AND PSEUDO-CLASSIFICATIONS BASED ON EXTERNAL RELEVANCE

by

David M. Jackson

Technical Report No. 69-3

on work performed under

Grant No. GN-534.1, National Science Foundation

April, 1969

Computer and Information Science Research Center
The Ohio State University
Columbus, Ohio 43210

ABSTRACT

The idea of pseudo-classification based on external relevance is introduced and compared with the more usual classifications derived by associative techniques. A general model for an information retrieval system using term classification is described. The derivation of a set of operators, or perturbations, for adjusting pseudo-classifications and preventing their deterioration is given for a particular match function conforming with this model. The use of pseudo-classifications both for the prediction of relevant documents and for the evaluation of retrieval systems with respect to their theoretical optimum is discussed. The concept of the improvability of a retrieval model with respect to its constituent submodels is introduced and elaborated upon.

PREFACE

This report is the result of research conducted on classification techniques for informational retrieval systems supported in part by Grant Number GN-534.1 from the Office of Scientific Information Service of the National Science Foundation to the Computer and Information Science Research Center, The Ohio State University.

The research was administered and monitored by The Ohio State University Research Foundation as Project RF 2218.

ACKNOWLEDGMENT

I should like to acknowledge the continued help and useful criticism of Dr. R. M. Needham of the Cambridge University Mathematical Laboratory, England, during the course of this work.

TABLE OF CONTENTS

SECTION I

1. INTRODUCTION

1.1 The Use of Relevance in Evaluative Retrieval System

In recent years a number of experiments have been performed to examine the application of term classifications to information retrieval. It seems evident from the work of Salton (1), and of Sparck Jones and Jackson (2) that a certain measure of improvement in performance[1] over simple term retrieval may be obtained by using classifications generated automatically. Both groups have established independently that small tightly structured classes of terms constitute a classification favourable to retrieval applications, but it is as yet unknown whether the improvement gained by using such classifications may be increased still further. The classes used in these experiments have in the main been generated automatically by making use of the co-occurrence of terms in documents. Doyle (3) has discussed the question of using co-occurrence as a measure of similarity and has pointed out a number of difficulties which this raises. It is, however, not at all easy to see how his criticisms can be met in practice without the danger of introducing further difficulties.

---

[1]Performance throughout this paper relates only to the retrieval or non-retrieval of relevant or non-relevant documents. No account is taken of 'hardware factors.' Thus, for example, the amount of effort expended in extracting relevant documents is not taken into consideration.

In the experiments of Sparck Jones and Jackson (4), attention has been focused specifically on the effect of automatically generated classes on retrieval performance. Their studies have involved research both in classification theory and information retrieval theory, and have been directed towards finding classification and retrieval algorithms which result in improvements in performance beyond that obtained by simple keyword coordination. An assessment of the effect of two different classifications of the same document collection and the same set of requests on retrieval performance involves a comparison between the two recall-precision curves. This was achieved by 'rule of thumb' since the authors were interested principally in an overall improvement for all coordination levels. Thus, when the curves for two experiments crossed, improvement was regarded as uncertain. Salton and Lesk (5) have used statistical methods to establish whether a limited number of classifications and thesauri display consistent improvement over a number of different document collections. Another evaluation measure has been developed by Swets (6), who uses a decision theoretic approach.

The approaches outlines above are similar in a number of respects. First, both use the term descriptions of the documents in the collection to produce term co-occurrences, and thence a classification of terms. Secondly, both attempt to evaluate the performance of the retrieval procedure by using the base set of requests and a table of documents relevant to each request. Thirdly, neither attempts to adjust a classification on the basis of information gathered during evaluation. This is not intended as a criticism of these approaches, for both set out to establish whether classifications derived solely from co-occurrences of terms

within documents can improve performance. A subsequent adjustment of the classification would, therefore, be outside the terms of reference of these investigations. Finally, neither can give a clear indication of the best performance which theoretically may be obtained from their retrieval systems given the document collection, the base set of requests and the table of documents relevant to the base set of requests. Some work in this general direction, however, has been done by Cleverdon and Keen (7), who have examined the exhaustivity and specificity of several index languages. Their arguments are based on the assumption that the user who approaches the document collection will, in all probability, not be familiar with the keywords which are available for specifying his request. Indeed, in a system which is designed to serve the user, there is no reason why he should be. They have put into clear relief the problem of ascertaining the actual intention of the user when he formulates his request, and have called in question the meaning and interpretation of relevance. For example, are documents judged relevant on the basis of the actual request the user formulated or are they judged on the basis of the request he should have formulated, had he complete knowledge of the term vocabulary and a clearer idea of his own request?

These remarks make it clear that there are four sources of information for an evaluative retrieval system, namely, the document collection, relevant to the base set of requests. The latter, together with a performance measure, enables the system to evaluate its performance by comparing lists of documents actually retrieved with lists of documents it should have retrieved. Not all of the four sources of information are independent of each other. Under the hypothesis that:

1.1.1 'co-occurrence of terms within documents is a suitable measure of the similarity between terms' a classification may be generated automatically. This hypothesis will be referred to as the Association Hypothesis. Alternatively, the classification may be produced manually. The dependence between the document collection, the base set of requests, the term classification and the table of relevant documents is, however, less tractable. Suppose, for example, that a match function, which measures the coefficient of matching between a document and a request, is designed, and that this coefficient varies with the coordination level of a match both on terms and on classes. Then with the hypothesis, which will be called the Relevance Hypothesis, that:

1.1.2 'coordination is positively correlated with external relevance' (i.e., that relevance may be defined algorithmically) we could imagine a classification and a match function which retrieved all and only relevant documents, namely those with high match coefficients. In practice, however, this position is seldom attained; partly because the match function is an imperfect approximation to that function which actually corresponds to external relevance, if indeed such exists; partly because the classification is defective; partly because there may, in fact, not be sufficient information provided to the system to achieve the best[2] performance; and partly because the notion of relevance may not be well formed and may result in inconsistencies of some sort. The difficulty is that we are attempting to simulate the external judging of relevance

---

[2]Best performance is used here in the sense of complete agreement with the document-request relevance table.

by using an assortment of classification algorithms based on the association hypothesis and an assortment of match functions based on the relevance hypothesis. Moreover, the simulation is undirected since we have no guide as to how a better simulation may be achieved. It may be that the lack of success in the field is attributable to the fact that the approach to retrieval we are using is overdetermined. We have attempted to inject more information into the system by introducing hypotheses rather than by utilizing the information available in a more economical fashion. It is the utilization of the available information in a more economical way and the ideas which this leads to which this paper is to explore.

1.2 The Use of Relevance in Pseudo-Classifications

The relationships which hold between the categories of information used in an evaluative retrieval system were described briefly above. To make descriptions of this type more precise it is convenient to introduce the notion of model. Suppose that a set of experiments is designed to examine a particular aspect of information retrieval. Functions and processes are designed according to a set of rules and these, when fused together, form the retrieval system. The set of rules which govern the construction of each part of the system represent a description of a particular approach to the solution of information retrieval. These rules themselves may be, in effect, hypotheses within the field of information retrieval and an evaluative retrieval system allows these hypotheses to be tested. A complete knowledge of the set of rules gives, ideally, complete knowledge of a particular approach to information retrieval. The

set of rules is called a model. The set of rules may be such that a number of quite different systems can be constructed to conform to them and each such system is called a representation of the model. Within a model of information retrieval there will be models, or submodels, for each of the logically distinct processes which together comprise the system. Thus, for example, there will be a classification model if classification is required as an adjunct of the retrieval system. There will be an information model describing the categories of information required by the system, together with statements of the assumptions made about these categories. There will be a relevance model containing statements about the way in which external relevance is simulated by the system. Finally, there may be an evaluation model containing statements of the assumptions made about the evaluation of the output of the retrieval system. Once the model has been defined it may be found that experiments can be designed to test the validity of the complete model, that is all representations within the model, rather than the validity of a single representation.

Section 1.1 contained an informal specification of part of a particular model of information retrieval in its reference to the relevance and association hypotheses (1.1.1 and 1.1.2) and the interrelation of the four categories of information listed in that section. In this model, the classification and match function are used together with the document descriptions to simulate the external judgments of relevance for the base set of requests. The information contained in the relevance table is disregarded in the construction of the system and is used post facto, evaluatively, to corroborate or contradict the result

of the classification algorithm or the match function. Instead, suppose that the requests are well-formed, in that they genuinely represent the requests which the user intended to submit to the system, and that the table of relevance appropriate to those requests is incontrovertible. The match function may be defective but is assumed to be at least a distant approximation to our intuitive notion of relevance. Each request is taken in turn and a classification of terms is gradually constructed which will confer a high match coefficient on those documents which are judged relevant. Each constructive step consists of an operation altering the membership of selected terms to selected classes of the classification. Furthermore, it is ensured that a subsequent alteration of the classification to accommodate the relevance judgments of another request does not affect the classification in a way which would be detrimental to the requests already examined. The classification so constructed deserves the name only insofar as it consists, de facto, of classes of terms. There may be little reason to assume that such a classification would consist of classes of terms which would represent generic concepts associated with the document collection. Subsequent analysis, for example, by comparison with classifications generated from term co-occurrences, might demonstrate that this is so. In the absence of this, however, we choose to draw a distinction between these two classifications by referring to the former as a pseudo-classification, and it is to be regarded, for the moment at least, as a purely formal construction. Provided such a classification may constructed, we are in a strong position. By construction, performance for the base set of requests will be the best attainable for all representations of the

classification model, given the particular model of retrieval. A subsequent request, one which does not belong to the base set, will be examined by the system and will be retrieved providing the match coefficient is sufficiently high. Of course, there is no absolute guarantee that the documents retrieved in this way will be judged relevant but the same criticism may be leveled at any other system. This point is elaborated in the next section.

SECTION II

2. THE USE OF PSEUDO-CLASSIFICATIONS IN RETRIEVAL

2.1 The Predictive Use of Pseudo-Classifications

The procedure which has been outlined constructs pseudo-classifications, which produce the best performance when used in conjunction with the apparatus of information retrieval. A distinction has been drawn between a pseudo-classification and a classification, for the former is a pure construction, a deus ex machina, while the latter is, as far as current classification research permits, a more fundamental grouping based on the resemblances between terms. The problem arises of predicting the effect of a pseudo-classification on retrieval performance when a new set of requests, distinct from the set used in constructing the pseudo-classification, is offered to the system. There is formally no guarantee that the system will respond in anything but a perverse way, for it must be clearly remembered that the pseudo-classification is derived from a particular set of requests and a particular set of relevance judgments over the whole document collection. We hope that the procedure has somehow generalized the notion of relevance from a number of specific instances of relevance of documents to requests and has enclosed it in the pseudo-classification in such a way that the system may predict which documents bear the same relationship to subsequently presented requests as certain prescribed documents bear

to the base set of requests. It is as if we have tried to distil relevance itself from repeated analogical statements such as "the relationship between D1 and R1 is the same as the relationship between D2 and R2" where it is known that document D1 is relevant to request R1 and document D2 is relevant to request R2. Although it is hoped that generalization has taken place, this cannot be assumed as the following hypothetical set of requests and relevance judgments demonstrates. Suppose that each request is disjoint from all other requests, each request has one document relevant to it and no document is relevant to more than one request of the base set of requests. A possible pseudo-classification for this configuration would be represented by the assignment of each term to a class by itself, and such a configuration obviously would not extend to an arbitrary request. Although experimental demonstration is required to show that this is unlikely, there are a number of arguments which might constitute a plausible defence against this eventuality. The first point is that the match function is designed to reflect an intuitive notion of relevance as formulated in the relevance hypothesis (1.1.2). The classification is constructed by operations involving the assignment of terms to existing classes and the creation of new classes by the assignment of terms to them not yet gathered together into a class. Pairs of terms within a class are used in retrieval as if they are intersubstitutable and it is the intersubstitutable pairs which yield good retrieval performance which we hope to locate. Although there may be some choice available in the selection of which terms and classes to operate upon and although these terms may be brought together into such classes randomly in the event of no other

basis for decision, it is not unreasonable to suppose that, as the classification develops it will become more determinate in that the opportunity for random assignments will diminish since the number of assignments to be made is finite. Nor is it unreasonable to suppose that the later assignments of terms to classes may have the effect of diminishing the possibly deleterious effect of the random assignments made during the early stages of the construction of the classification providing we allow terms to be removed from classes. In addition, we may imagine that a large sample of requests is taken as the base set of requests. Providing the sample is large enough, it is unlikely that a new request, in all probability similar to one in the base set, would produce a radically different set of documents on retrieval. However, this does raise the question of what constitutes a sufficiently large sample of requests. The problem of extending the system to deal with new requests will be referred to as the problem of generalizing.

The only satisfactory solution is to choose a base set of requests in such a way that the set is representative of all future requests which may be submitted to the system. There are a number of ways of doing this. Suppose, for example, that it were possible to make a statement about the distribution of terms in the requests which may be submitted to the system. In this case probabilities could be assigned to the output of the system to indicate that, although a document may be deduced relevant to a request, it is only guaranteed relevant by the system with the specified probability. Note that this is not the same as degree of relevance since the document may be entirely irrelevant to the request. This is a little unsatisfactory in the light

of the system's anticipated good response to requests of the base set.

Another possibility is to require that each request satisfy a condition which would be phrased as generally as possible, and the pseudo-classification would be constructed only with requests satisfying this. The system in turn would be required to respond only to requests which were of this type. Such a condition may contain a statement about the co-occurrence of terms within requests and we may, therefore, find ourselves in an inconsistent position. We attempt to base a retrieval system on considerations which do not involve classification techniques per se, but we require some kind of classification in order to ensure that the procedure will be capable of generalizing to subsequent requests.

A final remark in this connection is that there may be available a large collection or requests and the associated relevance judgments, and that although it cannot be demonstrated categorically, they are highly likely to represent a typical and representative sample of all the requests which may be put to the system in future, in that all the different kinds of request are adequately represented. Although this remark is pragmatic and formally indefensible, it may happen in a practical application that this state of affairs nevertheless obtains.

2.2 The Evaluative Use of Pseudo-Classifications

We shall now consider the use of which pseudo-classifications may be put in evaluating retrieval performance. Consider the following experiment. An information retrieval system is designed which conforms to some model. A match function is specified which is intended to measure the relevance of a document to a request. The design of this function

is based on intuitive ideas about relevance and is an internal analogue of relevance as judged by the users of the system. The function may be good or bad, according to its success in retrieving relevant and only relevant documents, and may be subject to change and modification as our ideas about the internal representation of relevance change. We also possess a number of algorithms for producing term classifications conforming to a model of classification. We want to isolate the classification which, when used in conjunction with the match function, results in highest retrieval performance. These then are the pieces of apparatus required for the experiment. There remains, however, the problem of evaluation. Although the use of precision-recall curves for this purpose seems to be well established in the literature, there is still criticism of them and they should, therefore, be regarded only as a temporary solution. There is no one method which is generally accepted. For example, Cleverdon and Keen (7) computes recall and precision without regard to the order in which documents are retrieved. Salton (1), on the other hand, is interested in determining whether the relevant documents are retrieved first. Swets (6) proposes a decision theoretic approach.

Suppose now that a certain match function is decided upon for the experiment. Classifications are produced by a number of algorithms and we wish to find which of these yield the best retrieval performance for a particular way of measuring performance. Each of these classifications, in turn, is used in the retrieval system, and the resulting performances are compared in a simple way to determine the best. With luck the results may suggest a variation in classification technique and by experimenting

of this kind it is hoped eventually to arrive at a classification algorithm which will give the best performance.

There is an obvious difficulty in this approach. For a given document collection, base set of requests and table of relevance there is a best possible level of performance, according to some measure of performance, the given match function and the classification model. The measure of performance customarily used do not relate the performance of a retrieval system to the best theoretical performance for the retrieval model, with the result that there is no indication of the extent to which the system may be improved or the direction in which such improvements may occur. Suppose, however, that it were possible to examine all the classifications in turn which satisfied the classification model in a retrieval system with a given match function and a given measure of retrieval performance. The maximum theoretical performance for the model would be given by the classification which resulted in performance better than any of the other classifications, since all possible representations in the model would have been examined in order to make this assertion. Such a classification is a pseudo-classification since it is the one which agrees most closely in retrieval with the relevance table for the base set of requests. Moreover, the pseudo-classification does not depend upon the measure of retrieval performance but more directly on the extent to which the relevance judgments set out in the relevance table have been reproduced by the system. The effect of the pseudo-classification may be assessed subsequently by choosing a particular measure of retrieval performance which uses this information,

and the numerical values which result will represent the best possible performance obtainable with the given models.

The problem, however, cannot be approached in this way for although the number of partitions, and, therefore, the number of classifications, of a finite number of terms is itself finite, the amount of computer time needed for the evaluation of them all would be prohibitively large. Instead of enumerating them all, we need a method for isolating those which are the best for the particular retrieval experiment. The point here is that we are no longer interested in the extraction of classifications in the sense of coherent groups of terms. We are only interested in formal groups of terms which result in good retrieval performance because the hypothesis that a particular classification technique is the best for a specific retrieval experiment is precisely the hypothesis which we mean to test. These partitions are the pseudo-classifications, an algorithm for the construction of which has been alluded to earlier in Section 1.2.

The following way of measuring retrieval performance is suggested by the foregoing remarks. A particular match function and performance measure are chosen. A pseudo-classification is constructed (by algorithm rather than by selection from a complete enumeration of the representations of the classification model) which, when used in retrieval, gives rise to a particular level of performance. This is the theoretical best for the data, for the classification and retrieval models, and for the match function. The proposed measure of performance measures the departure of the performance in a particular case from this theoretical best. If this departure is small, little improvement may be

gained by changing the classification technique and it would indicate that further experimenting in this direction would not be profitable. The introduction of another match function, however, would result in a different pseudo-classification and a different estimate of the theoretically best performance. If this were better than the optimum for the previous match function, then the choice of match function may be regarded as more useful in retrieval than the previous one in that it permits the system, at least theoretically, to achieve an improvement in performance. If the measure of retrieval performance were changed it would not be necessary to regenerate the pseudo-classification since this does not depend on the formulation of the measure. It is in this area that pseudo-classifications are particularly applicable; that is, in an evaluative rather than a predictive role. The following definition is, therefore, made:

2.2.1 The difference in the performance in retrieval of the pseudo-classification and the automatically generated classification (according to some measure of performance) gives an estimate of the <u>improvability</u> in performance of the retrieval model for the match function and classification algorithm which have been applied.

Although the use of this performance measure (2.2.1) may indicate whether it is more profitable at a particular stage of research to improve performance by changing either the classification algorithm or the match function no indication is given of how this is to be done. We are in the odd position of knowing which classification of the model gives the best performance yet being unable to supply a means of deriving it without recourse to the relevance judgments of the base set of requests.

In spite of this, its usefulness as an evaluative device is not impaired. It does not replace the researcher; it assists him to be more precise in his analysis of a particular experimental model.

2.3 The Isolation of Inconsistencies in Relevance Judgments

The practicability of constructing a pseudo-classification must now be raised. It may happen that after the classification has been constructed to give the correct response to a number of requests of the base set, a request is encountered whose processing conflicts diametrically with some of the conditions set up to prevent deterioration of the classification with respect to previously processed requests. It is possible that this results from an inconsistency in the judgment of relevance of documents to requests as supplied by experts in the field covered by the document collection. However, we expect that the assessment of external relevance by an individual does not lead to serious inconsistency and that the same is true for the determination of relevance by consensus of opinion. Detailed work by Resnick and Savage (8) on the consistency of human judgments of relevance support this view. It may also transpire that the number of conditions which have to be constructed to prevent deterioration, or the number of decisions which have to be taken and are represented by the assignments of terms to classes, increases at such a rate with the number of requests processed that the classification becomes overdetermined at an early stage of construction. It is hoped to show that this does not, in fact, happen. The best which can be done with a request which leads to an inconsistency is to reject it and tolerate a small decrease in performance over the base set of

requests. It would be of interest to determine the proportion of requests of this type to successfully processed requests and to examine the conditions which gave rise to the inconsistencies. The origins of the inconsistencies may, of course, be difficult to locate, but at least it will be known that a particular document-request pair cannot be manipulated in the pseudo-classification to satisfy the conditions set up for earlier pairs without impairing retrieval performance. This in itself is of interest as a guide to subsequent work on the construction of classification and match algorithms within the retrieval model.

On a purely practical level, there is little defence against inconsistent judgments of external relevance. If relevance judgments were to be completely idiosyncratic, a retrieval system would have to be constructed for each user, and the only way forward would be by interactive techniques. If relevance judgments were to be arbitrary, no system could function at a predictable level of performance. A fixed, that is, non-interactive system relies on a consensus of opinion of users as to the relevance of documents to requests.

SECTION III

3. A RETRIEVAL MODEL

3.1 The Information Model

An information model (vide 1.2) is a statement of the categories of information used to describe the system followed by a statement of the assumptions made about these categories. Within this model, algorithms may be designed to perform specified operations and they must not refer directly or indirectly to information which is outside the model.

Section 1.1 referred to the four sources of information required by an evaluative retrieval system. These are:

3.1.1 D the set of documents defined extensionally by terms. This is the document collection.

3.1.2 R the set of requests defined extensionally by terms. This is the base set of requests.

3.1.3 Z the set of requests defined extensionally by the documents relevant to them. This is the set of relevance judgments.

3.1.4 C the set of classes defined extensionally by terms. This is the term classification.

These are similar in that each defines one set of elements extensionally in terms of another set of elements. Each, therefore, may be regarded as a rectangular incidence array giving the occurrence of an element of

one type with an element of another type. The sign '_' indicates that we are considering a set of elements defined in terms of another set (i.e., an array) rather than a single element defined in terms of a set of elements. Attention is confined to documents indexed by using simple term coordination and document collections indexed probabilistically, for example, by the methods of Maron and Kuhns (9), are not considered. The reasons for this are threefold. First, the document collection used is not indexed probabilistically, although some attempt might be made to rectify this. Secondly, it is felt that tests with undifferentiated terms logically should precede any experimenting with term weighting in order first to establish a basis for comparison. Thirdly, the construction of pseudo-classifications is simplest in this case since a single operation on the classification involves the dichotomous choice between the removal or insertion of a term. The probabilistic case is more complex since the choice of a value to be assigned to the weight is no longer dichotomous.

The terms which specify the requests also appear unweighted as do, for quite different reasons, the documents relevant to the base set of requests. It may be possible to construct a scale of relevance and assign degrees of relevance of documents to requests according to this scale. Instead, those documents have been selected which have been given the highest degree of relevance. The scale of relevance, therefore, as applied externally, has two values, namely, relevant and not relevant. It is realized that the external judgment of relevance is more complex than this and that there is no simple division of documents into those relevant to a request and those not relevant to a request. The insertion of a third category would be more realistic, and such a

category, namely "of unspecified relevance" would serve to separate the polarities of relevance more clearly. The quantification of the degree of relevance of documents to requests has further difficulties of quite a different nature. Suppose, for example, that it were possible to place in order of increasing relevance to a particular request all the documents of the collection. As soon as numerical values are assigned to each document to quantify its degree of relevance, metric properties are assumed about the scale of relevance. During the course of retrieval, arithmetic operations on degrees of relevance are performed which tacitly assume the truth of statements like, for example: "document D1, whose degree of relevance to a request R is i is i/j times as relevant to R as document D2, whose degree of relevance to R is j." Until a metric has been established for degrees of relevance such statements remain indefensible. Resnick and Savage (8) have proposed to make a study of this question. For these reasons it is decided to work with categories of relevance rather than degrees of relevance; that is, with a qualitative scale rather than a quantitative scale. The two-valued scale has been adopted since the main body of data for testing purposes is reducible to this form.

3.2 The Classification Model

The model by which classifications to be constructed are constrained is as follows. Membership of terms to classes is a binary property; the object either belongs or does not belong to a class. The probabilistic assignment of terms to classes is excluded. All assignments of objects to classes as a priori independent and overlapping

classes are allowed and indeed expected. Finally, the classification is non-hierarchical. It should be noted that the classes of the classification are defined extensionally. Terms are not assigned to classes according to their satisfying a known condition on the class. It may, nevertheless, transpire that classes have useful properties in terms of the character of the vocabulary.

The model of retrieval, therefore, is such that all sets encountered are defined extensionally and non-probabilistically. For explanatory purposes we shall refer to 3.1.1, 3.1.2, 3.1.3, and 3.1.4 as the retrieval environment. This is slightly different from the way this term has been used elsewhere (Jackson (10)), but throughout the remainder of this paper it will be used consistently with this meaning.

3.3 The Relevance Model

The match functions which will be applied will only make use of the information contained in or derivable from the environment. We shall, therefore, write:

3.3.1 $$1 = M(D, R, f(D, \underline{C}), f(R, \underline{C}))$$

where 1 is the match coefficient corresponding to the match function M applied to an arbitrary document D in $\underline{D}$ and an arbitrary request R in $\underline{R}$ using the classification $\underline{C}$. f is a function which produces from a description of a set specified using terms a description using classes. $f(D, \underline{C})$ is equivalent to S(D, O, C) defined by Jackson (10). Formulations of f will be given in Sections 5.2 and 6.1. The purpose of f is to provide a means of recovering in class matches the term matches which were missed on simple matching of the term descriptions for R

and D because a term was used in one of these and a variant of this term in the other. The relationship between these terms may or may not be one of actual synonymy in natural language. The intention is that the classification should contain classes of terms which are mutually intersubstitutable and result in good retrieval performance.

In accordance with the relevance hypothesis (1.1.2) we require that M should be a monotonically increasing function of the number of terms or classes in common between R and D. Its behaviour with respect to the terms belonging to one of them but not to the other is not specified. This is a subject for precise formulation in a specific realization of M which satisfies the conditions mentioned. The modifications to the pseudo-classification as it is being constructed will be seen (vide 4.3) to involve the addition of terms to classes already defined or the grouping of terms to form new classes. The size of the classes and their number will, therefore, vary during construction and no particularly relevant interpretation may be put on them. In addition, if the match function is allowed to depend on them explicitly, the classification will be unalterable or will certainly deteriorate with respect to already processed requests as new requests are examined. The dependence of the match function on these two quantities is, therefore, explicitly proscribed.

The values resulting from applying the match function to a document which is absolutely relevant to a request and to a document which is absolutely irrelevant to a request may be specified at will. Although absolute irrelevance is not as precise an intuitive concept as absolute relevance (for one can always find some reason for saying

that a document is slightly relevant to a request, whatever it is, in a collection of restricted subject matter), we shall rest content simply to regard it as being that relationship which exists between document and request and which obtains at the lowest value of the match coefficient. A corollary of the relevance hypothesis is that this coefficient must increase monotonically to its maximum value which represents absolute relevance. We shall, therefore, require that the bounds of the match function be finite and that these bounds are attained, at least in theory, by the function. The additional conditions on the match function are, therefore:

3.3.2 M is a monotonically increasing function of the number of terms or classes in common between R and D.

3.3.3 M is independent of the size of any class in C and of the number of classes in C.

3.3.4 1 is bounded above and below, and attains its bounds.

It might happen that once the classification has been constructed using the base set of requests that it is still underdetermined. Additional requests and their appropriate relevance judgments could be used to complete the classification only if the match function does not depend on the number of requests in the base set. Similarly, additional documents could only be added to the collection provided the match function does not depend on the size of the collection. Since both of these are valuable properties of a retrieval system we shall add the appropriate conditions on M:

3.3.5 M does not depend on the size of R.

3.3.6 M does not depend on the size of D, or on the size of the term vocabulary.

Two examples of match functions are given by Jackson (10).

So far, there is no criterion of relevance in the model. Any criterion is bound to be arbitrary to a certain extent, for it is never possible to have complete knowledge either of the document collection or of the mind of the user of the system. To use the complete document unprocessed is as far from the solution as hoping to provide a complete analysis of the document, revealing in complete detail the complexity of the structural and semantic relationships between all the linguistic elements in the document. We have to make do with approximations, which we hope will reveal the salient features of the collection for the purposes of automatic retrieval. In the model, the upper and lower bounds of the match coefficients which represent the polarities of relevance are known. We have attempted by 3.3.1 to establish inside the model a scale of relevance between these poles and somewhere along this scale we must define a value, above which we retrieve documents and below which we suppress them. This value is called the critical value of the match coefficient. Categories of relevance are assigned, therefore, to retrieved documents, rather than degrees of relevance. In the absence of any evidence to the contrary, it will be assumed that the values of the match coefficient are distributed over the document collection in such a way that the critical value is the value midway between the extreme of the match coefficient. Subsequent experiment may cause us to revise this assumption. Thus,

for deciding on a suitable value for the critical point in the scale of relevance, the following hypothesis is made:

3.3.7 The critical value is the arithmetic average of the upper and lower extrema of the match coefficient.

This will be referred to as the critical value hypothesis.

The following notation is introduced in connection with the above assumptions. Suppose that l^+ and l^- are the upper and lower bounds of the match coefficient l, respectively, and suppose that l_0 is the critical value of l. In accordance with 3.3.7, l_0 is defined as:

3.3.8 $$l_0 = \tfrac{1}{2}(l^+ + l^-) .$$

We shall use the binary asymmetrical relation @ and $\not@$ to denote 'relevant to' and 'not relevant to', respectively. Thus:

3.3.9 $$l \geq l_0 \iff D \mathbin{@} R$$

and $$l < l_0 \iff D \mathbin{\not@} R .$$

For completeness, we also introduce the binary asymmetrical relations @* and $\not@^*$ to denote 'absolutely relevant to' and 'absolutely irrelevant to', respectively. Thus:

3.3.10 $$l = l^+ \iff D \mathbin{@^*} R$$

and $$l = l^- \iff D \mathbin{\not@^*} R .$$

SECTION IV

4. METHODS FOR CONSTRUCTING PSEUDO-CLASSIFICATIONS

4.1 Two Approaches

Within the model of retrieval set out in Section 3 two approaches in the construction of pseudo-classifications may be distinguished. The first approach may be regarded as an attempt at an analytic solution by determining the 'inverse' of the match function. This is seen more clearly from 3.3.1 which, in the context of Section 3.3, defines the match coefficient in terms of other elements of the model. This relationship is open to more general interpretation in which it is regarded as an equation connecting 1, D, R, and C by the functions f and M, which are given. Providing only requests of the base set are considered the values of 1, or at least their magnitudes with respect to the critical value of the match coefficient, are known. In the construction of pseudo-classifications, C is unknown so that 3.3.1 may now be interpreted as an implicit definition of C. Such a solution, however, depends on whether M possesses the requisite algebraic properties which enable the inversion to be performed. Even if these properties were known, the method would probably reduce to an amount of matrix manipulation--for D, R may be regarded as rectangular binary matrices--which would make such a solution uneconomic for the scale of document collection we hope to process eventually.

An alternative solution is by the method of perturbations, which has the advantage that less analysis of the match function is required. The classification is subjected to a number of alterations involving the insertion and deletion of terms from classes until the appropriate response to the base set of requests is elicited from the system. The alterations are carried out by perturbation functions. The difficulty with applying perturbations to the classification is to ensure that the method converges to a solution. Not only must convergence be proved but it must also be shown that the limit of this process is attained and is the required classification. For the moment we are content to outline a method which at least converges.

4.2 General Outline of a Method

Suppose that it were possible, by a suitable perturbation, to cause the system to give the correct response, relevant or not relevant, to a specific document-request pair. With a suitable convergence theorem it would be possible to treat each document-request pair independently of the others. The response for one document-request pair might be destroyed or impaired by later adjustments to accommodate other pairs but at worst it would be necessary to examine each pair a number of times. The convergence theorem would ensure that although a number of responses may be impaired, gradual convergence would nevertheness set in. Without such a theorem, it is necessary to use a method in which each document-request pair is examined once only and in which conditions are set up to prevent a response from being obliterated by the processing of subsequent pairs. The degrading of responses by adjustment of the classification to accommodate subsequent document-

request pairs will be referred to as deterioration. The conditions necessary to prevent the destruction of a response are referred to as deterioration conditions.

It is now possible to give an outline of a method for constructing pseudo-classifications:

4.2.1 i. Let C' be the state of the pseudo-classification.

ii. Perturbations are applied to C' until the correct response is given by the system to a particular (D, R). The state of the pseudo-classification is then C".

iii. Deterioration conditions are set up for (D, R) with respect to C".

iv. The process is repeated for the next (D, R).

4.3 Design of Perturbation Functions

Deterioration conditions for the method described in 4.2.1 are effective only if all the perturbations which might lead to deterioration have been examined. It is, therefore, necessary that the perturbations which affect the match function should be exhaustively enumerated and that there should be few enough of them to make the method practicable. For each document- request pair, all perturbations which might lead to deterioration must be examined. We shall require, as a purely practical constraint on the method, that the number of deteriorating perturbations be less than the number of document-request pairs to be considered. The algorithm for constructing pseudo-classifications is, therefore, less than order two in the number of document-request pairs.

The operations which are appropriate to altering the classification are:

4.3.1 Assignment of terms to classes.

$a(\{x_i\}^p \rightarrow \{y_i\}^p)$; < condition on $\{x\}^p$, $\{y\}^p$ > for all $i \leq p$

4.3.2 Removal of terms from classes and their assignment to other classes.

$r(\{x_i\}^p : \{y_i\}^p \rightarrow \{z_i\}^p)$; < condition on $\{x\}^p$, $\{y\}^p$, $\{z\}^p$ > for all $i \leq p$

where $\{x\}^p$ is an ordered set of p terms and $\{y\}^p$, $\{z\}^p$ are ordered sets of p classes, and where the conditions limit the choice of operands for a and r. x_i, y_i, z_i are the i-th members of the sets $\{x\}^p$, $\{y\}^p$, $\{z\}^p$, respectively. p is called the step length of perturbation. A complete enumeration of the perturbations with step length P which affect the classification involves the enumeration of all perturbations of smaller step lengths (i.e., P-1, P-2, . . ., 2, 1).

Another possible perturbation is the simple removal of p terms from p classes. This, however, might result in the complete evacuation of terms from the classification. The removal of terms from classes, however, is provided for in 4.3.2 and since in this case terms are reassigned to classes, there is no possibility of complete evacuation of the classification. For these reasons the simple removal of terms from classes without reassignment is not considered.

The effect of 4.3.1 on the classification is to classify terms (i.e., to insert terms into classes of the pseudo-classification) while the effect of 4.3.2 is to reclassify terms (i.e., to redistribute

terms among the classes of the pseudo-classification). A further distinction may be drawn. The classes mentioned in 4.3.1 and 4.3.2 are of two kinds. A class may already exist in the classification so that the effect of 4.3.1 and 4.3.2 is to add or remove terms. The effect, therefore, is to alter the consitution of a class. The number of classes in the classification remains unchanged or decreases. The classes may, however, be new in the sense that it is only when the functions have been applied that the classes enter the classification. They are not modifications of already existing classes for the number of classes in the classification increases. The effect of 4.3.1 and 4.3.2 when applied to such classes is to <u>create</u> new classes within the pseudo-classification. These distinctions are made use of in Section 4.5 where a method is proposed for selecting the appropriate perturbation to apply to a given document-request pair.

The choice of perturbation is further determined by the response which must be simulated for a given document-request pair. Suppose, for example, that the response of the retrieval system to the pair (D, R) is 'not-relevant'. Suppose further that 'relevant' is the correct response. A perturbation must be applied to the classification which has the effect of increasing the value of the match coefficient for (D, R) to a level greater than the critical value. To facilitate this selection it is, therefore, important that the perturbations should be grouped into those which increase the match coefficient, those which leave the match coefficient unchanged and those which decrease the match coefficient. The following terms are, therefore, used:

4.3.3 Increasing perturbations have the effect of increasing the match coefficient for a given document-request pair.

Level perturbations have the effect of leaving unchanged the match coefficient for a given document-request pair.

Decreasing perturbations have the effect of decreasing the match coefficient for a given document-request pair.

Once perturbation functions have been grouped according to their effects on the match function, it is clear how the deterioration conditions should be determined. If it is necessary to apply an increasing perturbation to accommodate a particular document-request pair, then conditions must be set up to inhibit the application of the decreasing perturbations which may lower the match coefficient below the critical value. Similarly, if decreasing perturbation is applied then conditions must be set up to inhibit the action of increasing perturbation functions.

The number of perturbations of the form 4.3.1 and 4.3.2 increases with the step length p. In order to satisfy the practical constraints on the model only single step (p = 1) perturbations will be examined in detail.

4.4 Deterioration Conditions

Suppose that t_i is a term and that C_j is a class in the classification. It follows from the model of classification defined in Section 3.2 that either $t_i \in C_j$ or $t_i \notin C_j$ and that membership is non-probabilistic. The classification $\underline{C}$ may, therefore, be represented by a binary array indicating the incidence of terms in classes. Thus,

4.4.1 $C_{ij} = 1 \Leftrightarrow t_i \in C_j$

$C_{ij} = -1 \Leftrightarrow t_i \notin C_j$

where $\underline{C}$ is used to denote both the classification and the array. During the construction of the classification, however, it is useful to use another possibility of membership of terms to classes. The value $C_{ij} = 0$ is to imply that on the basis of the information used so far, no decision may be taken about the membership of t_i to C_j although at a later stage of construction a definite choice may be made. During the construction of the classification the array is tri-valued. The values 1, 0, -1 are referred to as status values.

During the construction of the pseudo-classification, perturbations are applied which affect the membership of terms to classes and accordingly change the corresponding status values. The change in the status value is called a transition and is defined as:

4.4.2 $e_i \rightarrow e_j$ status value e_i is changed to status value e_j in the pseudo-classification.

The transition table T_{ij} gives the set of permitted transitions and is defined by:

4.4.3 $T_{ij} = 0 \Leftrightarrow e_i \rightarrow e_j$ not allowed

$T_{ij} = 1 \Leftrightarrow e_i \rightarrow e_j$ allowed

where $e_i, e_j \in (-1, 0, 1)$. From 4.3.1 and 4.3.2 the single step perturbations are of the form:

4.4.4 $a(x \rightarrow y); <$ conditions on $(x, y) >$

4.4.5 $r(x:y \rightarrow z)$; < condition on (x, y, z) > .

The deterioration condition associated with 4.4.4 is 'x must never be assigned to y'. This is achieved by forbidding the membership of term x to class y and by forbidding any transition from the status value -1. The effect on the pseudo-classification is, therefore:

$$C_{ij} = -1 \quad \text{where} \quad x = t_i \text{ and } y = C_j$$

and the required values in the transition table to ensure that this is never revoked are:

$$T_{-1,1} = 0 \quad \text{and} \quad T_{-1,0} = 0 \quad .$$

The change to the classification is effected if

$$T_{0,-1} = 1 \quad .$$

The deterioration condition associated with 4.4.5 is 'if x is in y then x may not be assigned to z'. This condition is recorded in the <u>condition table</u> $Q(k:i, j)$ defined by:

4.4.6 $Q(k:i, j) = 0 \iff r(t_k : C_i \rightarrow C_j)$ allowed.

$Q(k:i, j) = 1 \iff r(t_k : C_i \rightarrow C_j)$ not allowed.

It is important to note that a condition of this sort may not be revoked in a later stage of the construction of the pseudo-classification. That is, within the condition table the change <u>from</u> 1 to 0 is not allowed.

The effect of 4.4.4 on the pseudo-classification is:

$$C_{ij} = 1 \quad \text{where} \quad x = t_i \quad \text{and} \quad y = C_j$$

and for this to be possible the transition $0 \rightarrow 1$ must be allowed:

$$T_{0,1} = 1 \quad .$$

The effect of 4.4.5 on the pseudo-classification is:

$$C_{ki} = 0,\ C_{kj} = 1 \quad \text{where} \quad x = t_k,\ y = C_i,\ z = C_j$$

and for this to be possible the transition $1 \rightarrow 0$ must be allowed:

$$T_{1,0} = 1 \quad .$$

The remaining transitions $e_i \rightarrow e_i$ for $e_i \in (-1, 0, 1)$ and $1 \rightarrow -1$ are allowed since they are not explicitly excluded by the above analysis. These results are collated below. The transition table is, therefore,

4.4.7

		i = -1	i = 0	i = 1
j	-1	1	1	1
	0	0	1	1
	1	0	1	1

The effect of 4.4.4 on the classification array is:

4.4.8 $\quad C_{ij} = 1 \quad .$

The effect of 4.4.5 on the classification array is:

4.4.9 $\quad C_{ki} = 0,\ C_{kj} = 1 \quad .$

The deterioration condition for 4.4.4 is:

4.4.10 $C_{ij} = -1$.

The deterioration condition for 4.4.5 is:

4.4.11 $Q(k:i, j) = 1$.

Two functions a' and r' are now introduced which have the effect of setting up the deterioration conditions 4.4.10 and 4.4.11 associated with a and r, respectively. The functions are called <u>conditional perturbations</u> and are defined as:

4.4.12 $a'(x \rightarrow y)$; < condition on (x, y) >

whose effect is $C_{ij} = -1$ where $x = t_i$ and $y = C_j$ (vide 4.4.10)

4.4.13 $r'(x:y \rightarrow z)$; < condition on (x, y, z) >

whose effect is $Q(k:i, j) = 1$ where $x = t_k$, $y = C_i$, $z = C_j$ (vide 4.4.11).

4.5 Precedence of Perturbations

It has been seen in 4.3 that perturbations may be grouped in two different ways; according to their effect on the match coefficient and according to their general effect on the classification. The selection of an appropriate perturbation to apply for a given document-request pair is determined to a certain extent by a knowledge of the system's response and the response specified in the relevance table. If, for example, the match coefficient for a document-request pair is lower than the critical value and it is known that the document is relevant to the

request then an increasing perturbation function is required. The choice of perturbation function is further determined by establishing a precedence between perturbations according to their effect on the pseudo-classification namely classifying, reclassifying or creating (vide 4.3). These three groups of perturbations will be denoted by c, r, and n, respectively. This is not, however, an exclusive grouping since a particular perturbation may be, for example, both r-type and n-type, that is its effect is both to reclassify and to introduce a new class. A complete exclusive grouping consists of the group c (classify), r (reclassify), cn (classify and create new class) and cr (classify and create new class). The type of a perturbation is, therefore, defined as:

4.5.1 type = c or r or cn or rn .

The first precedence to consider is that between r and c. During the construction of the pseudo-classification it is advantageous first to attempt to reclassify the terms already classified until no further reclassification is possible. At this point more terms are admitted to the classification and are subsequently reclassified as appropriate. The precedence rule which achieves this effect is:

$$r > c \quad .$$

The same argument hold for rn and cn and we obtain the rule:

$$rn > cn \quad .$$

A final requirement is that the classification should be constructed with as few classes as possible to achieve the required distinctions among the terms. This position is approached, at least in principle, if the perturbations involving the creation of new classes have the lowest precedence. Accordingly, the precedence of the types of perturbation is given by:

4.5.2 precedence = r > c > rn > cn .

The selection of perturbations according to precedence and according to their effects on the match function does not lead to a unique function. There may be a number perturbations which fulfil the conditions and for each of these there may be a number of possible choices of operands satisfying < condition on (x, y) > or < condition on (x, y, z) > (vide 4.3.1 and 4.3.2). It will be seen that the perturbations change the match coefficient by an amount which is independent of the choice of arguments from among those which satisfy the appropriate conditions. Therefore:

4.5.3 the arguments for the perturbation are chosen at random from the class of suitable operands.

To prevent the classification from becoming overdetermined at an early stage of construction (in the sense described in Section 2.3):

4.5.4 the smallest number of perturbations are selected which together produce the required change in the match coefficient.

SECTION V

5. ENUMERATION OF PERTURBATIONS FOR A GENERAL MATCH FUNCTION

5.1 Scope of the Chapter

The enumeration to be given is of general applicability to the class of match functions to be defined. The perturbations to be deduced are predominantly concerned with the specific method of constructing pseudo-classifications which was outlined in 4.2.1, since the enumeration to be given will be complete as is required by that method. However, the perturbations deduced are relevant to any procedure for constructing pseudo-classifications which may be regarded as a 'method of perturbations' as defined in Section 4.1. A match function will be taken as an example and a complete list of the perturbations which affect it will be given together with their classification into r-type, n-type, c-type, increasing, level, and decreasing.

5.2 Definition of the Function f

The purpose of f, described in Section 3.3, is to produce a description of a request or document in terms of classes from the original term description. In contrast with term matching, in which a match is located when a term is found in common between document and request, a class match is located when a term from a request and a term from a document are found to belong to the same class. The function f facilitates the counting of class matches by producing the

appropriate set of classes in which matches may be sought. Such a set[1] of classes is called a class-description of the document or request. It may be useful to permit the production of the class-description from a subset of terms of the term-description. For example, since terms in common between document and request necessarily lead to classes in common between the corresponding class-descriptions, it may be appropriate, as far as the calculation of class matches is concerned, to experiment with the residue of the term-descriptions of the document and the request after matching terms have been removed. Accordingly, f is defined as follows.

Suppose that A is a set of terms and that $A' \subseteq A$. Suppose also that $\underline{C}$ is a classification of the terms and that U is a class within the classification. Then

5.2.1 $\quad t \in A' \wedge U \Leftrightarrow U \in f(A, \underline{C})$

f(A, $\underline{C}$) is called the class-description of A with respect to the classification $\underline{C}$.

5.3 Specification of the Class of Match Functions

Suppose that R and D are term-descriptions of an arbitary request and document and that R' and D' are their respective class-descriptions. Then,

---

[1]For explanatory purposes we prefer to use the word class to refer exclusively to groups of terms in the classification. Thus, the classification is an organization of terms into classes. The word set is to refer only to incidental constructions of terms, grouped together but not necessarily forming classes in the classification. Thus, for example, we shall call a request R a set of terms, for there is no particular reason why these terms should constitute a class in the classification. R', the class-description of R, is a set of classes.

5.3.1 $R' = f(R, \underline{C})$ and $D' = f(D, \underline{C})$

for some f satisfying 5.2.1. Two-by-two contingency tables with the indicated marginal totals are now defined for R and D and for R' and D'.

5.3.2

	D	$\overline{D}$	totals
R	a	b	n
$\overline{R}$	c	d	N-n
totals	m	N-m	N

	D'	$\overline{D}$'	totals
R'	a'	b'	n'
$\overline{R'}$	c'	d'	N'-n'
totals	m'	N'-m'	N'

5.3.3 where $N = a + b + c + d$ is the number of terms in the vocabulary

and $N' = a' + b' + c' + d'$ is the number of classes in the classification.

Match functions of the form M(a, b, c, d, a', b', c', d') will be considered; this class of match function is consistent with 3.3.1. Not all the arguments, however, are permitted for it is possible to expose an implicit dependence on N and N' by eliminating one of a, b, c, d, and one of a', b', c', d' by using the expressions for N and N' given in 5.3.3. This is explicity excluded by 3.3.6 and 3.3.3. In document-request matching d and d' are the least informative variables for the measurement of the similarity between the two term-descriptions and between the two class-descriptions. Accordingly, only match functions of the form

5.3.4 $1 = M(a, b, c, a', b', c')$

will be considered.

5.4 Introductory definitions

The operation "remove" defined in 4.3.2 consists of two actions which may be separated and considered independently. First, a term may be taken from a class to which it belongs, and second, the same term may be placed in another class. The combination of these two operations is equivalent to the remove operation. These two operations will be called take and place and are performed by the t-function and the p-function, respectively. The taking and placing of terms with respect to the same class has a null effect on the pseudo-classification and is, therefore, expressly avoided. The t and p functions are defined as follows:

5.4.1 $t(x, y)$ has the effect $C_{ki} = 0$ on $\underline{C}$, where $x = t_k$ and $y = C_i$. x is a term and y is a class of the classification $\underline{C}$.

5.4.2 $p(x, z)$ has the effect $C_{kj} = 1$ on $\underline{C}$, where $x = t_k$ and $z = C_j$ where C_j is a class of the classification $\underline{C}$.

The relationships between the a and r functions and the t and p functions are:

5.4.3 $$a(x \rightarrow z) \equiv p(x, z)$$

5.4.4 $$r(x{:}y \rightarrow z) \equiv t(x, y)\, p(x, z)$$

where the evaluation is carried out from left to right.

In considering the effect of the t and p functions on the class-description B of a term-description A of a document or request, two cases are to be distinguished. Suppose that a term is taken from a class C which possesses no other terms common to A. The class-description

of A will be altered by the complete removal of the class C from it and this will have an effect on a', b', c'. When the term is subsequently placed in another class there will be a further change in a', b', c'. The combined effect of these two changes is the effect produced by the remove operation r. Suppose, however, a term is taken from a class which has another term in common with the term-description A. In this case there will be no change in the values of a', b', c' although the subsequent placing of the term in another class may produce a change. The terms of A and the classes of B must, therefore, be grouped according to the effect of the t-function on them. Two conditions are required; one is to test whether a specific term, if taken from the class y, will cause that class to be omitted from the class-description of A; the other is to test whether there will be no such change. These conditions[2] are, respectively:

5.4.5 $\quad L_1(y, A) \equiv (N(v|v\epsilon y \Lambda A) = 1)$

5.4.6 $\quad L_2(y, A) \equiv (N(v|v\epsilon y \Lambda A) \neq 1)$

The value[3] of 5.4.5 will be T if the class v and the set A of terms have only one term in common, namely $y \Lambda A$. The value of 5.4.6 is T if the class v and the set A of terms do not have one common term--there

---

[2]The notation (x|< condition on x >) means 'the set of all x which satisfy the < condition on x >. Conditions are separated by semicolons. N(x|< condition on x >) is the cardinal of the set so defined.

[3]The values T and F are the values true and false of boolean conditions.

may be none or several. These two conditions are related by $L_1(v, A) = \overline{L_2(v, A)}$ but it is convenient to preserve their separate identity. Corresponding to these two conditions are two versions of the t-function. The t-function may be applied to a term x and a class y which are related to the term-description A in such a way that if x is taken from y then y will vanish from the class-description B of A. The function which has this effect will be called the t_1-function. Alternatively, if x is taken from y there may be no such effect on the class-description B of A and the function responsible for this will be called the t_2-function. These two functions are defined by:

5.4.7 $\quad t_1(x, y) \equiv t(x, y); L_1(y, A)$

5.4.8 $\quad t_2(x, y) \equiv t(x, y); L_2(y, A)$.

Suppose that X is a subset of A and Y is a subset of B where B is the class-description of the term-description A with respect to the classification. Suppose also that x is a term belonging to the class y. X and Y are defined by:

5.4.9 For any term x belonging to X, there exists a class y belonging to Y such that if x is taken from the class y then y will vanish from B.

5.4.10 For any class y belonging Y, there exists a term x belonging to X such that if x is taken from y then y vanishes from B.

5.4.11 If the class y vanishes from B when x is taken from y, then x belongs to X and y belongs to Y.

X is called the <u>domain</u> in A of the t_1-function. Y is called the <u>range</u> in B of the t_1-function.

Suppose that X is the domain in A of the t_1-function and Y is the range in B of the t_1-function. Suppose that the states of the classification before and after the application of $t_1(x, y)$ are $\underline{C}$ and $\underline{C}'$, respectively. Then it follows from the definition of X and Y that X and Y have the properties:

5.4.9' $\quad x \in X \Rightarrow \exists y \in Y \cdot y \notin f(A, \underline{C}')$

5.4.10' $\quad y \in Y \Rightarrow \exists x \in X \cdot y \notin f(A, \underline{C}')$

5.4.11' $\quad y \notin f(A, \underline{C}') \Rightarrow x \in X,\ y \in Y$

where $X \subseteq A$, $Y \subseteq B$ and $B = f(A, \underline{C})$

and[4]

5.4.12 $\quad X = (\forall x)(\forall y)(x \mid x \in A \wedge y;\ y \in B;\ L_1(y, A)) \equiv \underline{\text{domain}}\ (t_1;\ A;\ B)$

5.4.13 $\quad Y = (\forall y)(y \mid y \in B;\ L_1(y, A)) \equiv \underline{\text{range}}\ (t_1;\ A;\ B)$.

The range in B and the domain in A of the t_2-function are defined analogously.

The domain in A of t_1, the domain in A of t_2, the range in B of t_1 and the range in B of t_2 provide the grouping of terms of A and classes of B according to the effects of the t-functions. The effects

---

[4] $\forall$ is the universal quantifier. Although (x|< condition on x >) i.e., 'the set of all x's which satisfy the < condition on x >' uses the quantifier $\forall$ implicitly, the quantifier may be written explicitly to display the role of x. Thus, there is no difference in meaning between (x|< condition on x >) and ($\forall$ x)(x|< condition on x >).

of the t-functions may now be stated explicitly:

5.4.14 $t_1(x, y)$ decreases the cardinal of the range of t_1 by one if and only if x is in the domain in A of t_1 and y is in the range in B of t_1, and has no effect otherwise.

5.4.15 $t_2(x, y)$ has no effect on the cardinal of the range in B of t_2 if x belongs to the domain in A of t_2 and y belongs to the range in B of t_2 .

The four sets are related in the following way. Let Y_1 = range $(t_1; A, B)$ and Y_2 = range $(t_2; A, B)$. Then,

5.4.16 Y_1 and Y_2 are disjoint and cover B. That is $Y_1 \wedge Y_2 = \emptyset$, $Y_1 \vee Y_2 = B$

since a. if Y_2 is non-empty, there is no term x belonging to A which, when taken from any class in Y_2, will cause this class to vanish from B,

and b. if Y_1 is non-empty, there is no class in Y_1 which will not vanish from B if a suitably chosen term is taken from it.

5.4.17 X_1 and X_2 are disjoint but cover A. That is, $X_1 \wedge X_2 \neq \emptyset$ and $X_1 \vee X_2 = A$ where X_1 = domain $(t_1; A, B)$ and X_2 = domain $(t_2; A, B)$,

since there may exist a term x belonging to X_1 which, when taken from a class in Y_1 causes that class to be removed from B but when removed from another class in B does not cause that class to vanish. Thus, x may belong to X_2.

5.5 Effect of t and p Functions on a Document-Request Pair

Suppose that the document-request pair (D, R) is considered in which D is a document and R is a request. D and R are expressed by term-descriptions. Suppose also that their class-descriptions are D' and R', respectively, where

5.5.1 $D' = f(D, \underline{C})$ and $R' = f(R, \underline{C})$

for a classification $\underline{C}$ and a function f satisfying 5.2.1. The terms of R and D and the classes of R' and D' are separated into sets defined as follows:

5.5.2 $$F = R \wedge D \quad , \quad G = D \wedge \overline{(R \wedge D)} \quad , \quad H = R \wedge \overline{(R \wedge D)}$$
$$F' = R' \wedge D' \quad , \quad G' = D' \wedge \overline{(R' \wedge D')} \quad , \quad H' = R' \wedge \overline{(R' \wedge D')}$$

Thus, F' contains the classes common to R' and D'. H' contains the residue of R' after the removal of terms common to D'. G' contains the residue of D' after the removal of terms common to R'. F'. G', H' are disjoint. Similar statements hold for F, G, H. From 5.3.2:

$$|F| = a, \ |G| = c, \ |H| = b \text{ and } |F'| = a',$$
$$|G'| = c', \ |H'| = b' \ .$$

The analysis described above will now be applied to the class-description R' of R. R' may be partitioned into two sets of classes, denoted by F'_1 and F'_2, which are defined as follows:

5.5.3 $$F'_1 = \underline{\text{range}} \ (t_1; R, F')$$

and

$$F'_2 = \underline{\text{range}} \ (t_2; R, F')$$

F'_1 and F'_2 are disjoint and cover F', by 5.4.16. In addition, F''_1 and F''_2 are defined as follows:

5.5.4 $$F''_1 = \underline{\text{domain}} \ (t_1; R, F')$$

and

$$F''_2 = \underline{\text{domain}} \ (t_2; R, F') \ .$$

Analogous statements hold for an exhaustive and disjoint partition of H' into H'_1 and H'_2 with H' replacing F' in 5.5.3 and 5.5.4 above. Similarly, F' in D' may be partitioned into the disjoint sets F'_3 and F'_4 which cover F'. These are defined by:

5.5.5 $$F'_3 = \underline{\text{range}}\ (t_1;\ D,\ F')$$

and

$$F'_4 = \underline{\text{range}}\ (t_2;\ D,\ F').$$

In addition, F''_3 and F''_4 are defined as follows:

5.5.6 $$F''_3 = \underline{\text{domain}}\ (t_1;\ D,\ F')$$

and

$$F''_4 = \underline{\text{domain}}\ (t_2;\ D,\ F').$$

Analogous statements hold for the exhaustive disjoint partition of G' into G'_1 and G'_2. With these preliminary remarks, the effect of the t-function on (D, R) will be examined.

Suppose that x belongs to F''_1. Then by 5.4.9', a class y can be found such that if x is removed from y then the cardinal of F' will be reduced by one. There is, however, another effect if x also belongs to F''_4. A class will remain in D' which matched with the class in F' which vanished with the removal of x, and this class will be unaffected by the removal. It will, therefore, become attached to G', with the result that the cardinal of G' will be increased by one. The cardinal of H' will be unaffected. If x does not belong to F''_4 but to F''_3 the cardinal of G' will be unaffected.

Suppose that x belongs to F''_3. Then by 5.4.9', a class y can be found such that if x is removed from y then the cardinal of F' will

be reduced by one. In addition, if x belongs to F''_2 a class will remain in R which matched with the class in F' which disappeared with the removal of x, and this class will be unaffected by the removal. It will, therefore, become attached to H' with the result that the cardinal of H' will be increased by one. The cardinal of G', however, will be unaffected. If x does not belong to F''_2 but to F''_1 the cardinal of H' will be unaffected.

For any term x in H''_1 a class y in H'_1 can be found such that if x is taken from y then the cardinal of H' is reduced by one. The cardinals of G' and F', however, remain unchanged.

For any term x in G''_1 a class y in G'_1 can be found such that if x is taken from y then the cardinal of G' is reduced by one. The cardinals of F' and H' are unaffected.

If a term x in F''_2 is taken from a class in F'_2, there will be no effect on the cardinals of F', G', H' if x also belongs to F''_4.

Similarly, there is no effect if a term in H''_2 is taken from a class in H''_2 or if a term in G''_2 is taken from a class in G'_2. These effects on a', b', c' are tabulated in 5.5.8, below.

5.5.7 The increments in a', b', c' due to the t-function are denoted by $\Delta_t a'$, $\Delta_t b'$, $\Delta_t c'$, respectively.

5.5.8 Table of the effects of t(x, y); $x \epsilon X$ on a', b', c'. The four domains F''_1, F''_2, H''_1, H''_2 (defined in 5.5.4) and F''_3, F''_4, G''_1, G''_2 (defined in 5.5.6) are considered.

X	$\Delta_t a'$	$\Delta_t b'$	$\Delta_t c'$
F''_{14}	-1	0	1
F''_{23}	-1	1	0
F''_{31}	-1	0	0
F''_{42}	0	0	0
H''_1	0	-1	0
H''_2	0	0	0
G''_1	0	0	-1
G''_2	0	0	0

where:

5.5.9 $$F''_{ij} = F''_i \Lambda F''_j \quad .$$

The effect of the p-function, defined in 5.4.2, will now be considered. Suppose that X is selected from one of F, G, H, and the Z is selected from one of F', G', H', V, W where

5.5.10 $$V \epsilon \underline{C} \Lambda \overline{(R' \vee D')} \quad , \quad W \notin \underline{C}$$

and V and W are sets of terms and where $\underline{C}$ is the set of classes which together constitute the classification. For each selection of X and each selection of Z, p(x, z) has an effect on a', b', c' which does not vary with x or z provided that x belongs to X and z belongs to Z. The effect on a', b', c' of the p-function for $z \epsilon W$ and $x \epsilon X$ is indistinguishable from the effect for $z \epsilon V$ and $x \epsilon X$ whatever the selection of X from F, G, H. W is retained, however, since it is a 'created' or 'new' class as defined in Section 4.3, whereas V is a class of the

classification which belongs neither to D' nor to R'. It is the distinction between a class of the classification and a 'new' class which led to the precedence rules established in Section 4.5. For each of the three possible choices of X there are five possible choices for Z and since these may be made independently, the total number of pairs (X, Z) for which the effects of p(x, z); $x \epsilon X$; $z \epsilon Z$ on a', b', c' are to be examined is fifteen. It will be seen in 5.5.12 below that some simplification may be carried out to reduce the total number of pairs to eleven. These account for all the possible single-step p-functions which may be applied to (D, R).

Suppose that a term x which belongs to H is placed in a class z belonging to F' or H'. There will be no change in a', b', c' since no classes will have been introduced or removed. The effect of such an operation is to change the distribution of the classes among F'_1, F'_2, H'_1, H'_2.

Suppose that a term x which belongs to H is placed in a class z belonging to G'. R' will now have an additional class z in common with D'. Therefore, the number of matching classes will increase by one. The matching class z will be shifted from G' to F', thereby decreasing the cardinal of G' by one.

Suppose that a term x which belongs to H is placed in a class z belonging to V or W. The cardinal of H' will be increased by one and there will be no change in the cardinals of H' and F'.

Analogous statements hold for a term x belonging to G which is placed in a class z belonging to F', G', H', V or W.

Suppose that a term x which belongs to F is placed in a class z which belongs to F'. No new class matches will be made and no new classes will be added. There will, therefore, be no change in a', b', c'.

Suppose that a term x which belongs to F is placed in a class z which belongs to G'. The class z is, therefore, inserted into both D' and R' and the number of class matches increases by one. G', however, loses the class z, so the cardinal of G' decreases by one.

Suppose that a term x which belongs to F is placed in a class z which belongs to H'. The class z is, therefore, inserted into both D' and R' and the number of class matches increases by one. H', however, loses the class z, so the cardinal of H' decreases by one.

Suppose that a term x which belongs to F is placed in a class z which belongs to V or W. A new class match on z will be introduced without changing H' or G'. The cardinal of F' will, therefore, increase by one and the cardinals of H' and G' will be unchanged. The results are tabulated in 5.5.12 below.

5.5.11 The increments in a', b', c' due to the p-function are denoted by $\Delta_p a'$, $\Delta_p b'$, $\Delta_p c'$, respectively.

5.5.12 Table of the effects of p(x, z); $x \in X$; $z \in Z$ on a', b', c' for all relevant (X, Z).

X	Z	$\Delta_p a'$	$\Delta_p b'$	$\Delta_p c'$
H	R'	0	0	0
H	G'	1	0	-1
H	V or W	0	1	0
G	H'	1	-1	0
G	D'	0	0	0
G	V or W	0	0	1
F	F'	0	0	0
F	G'	1	0	-1
F	H'	1	-1	0
F	V or W	1	0	0

Now the effect of p(x, z); $x \epsilon H$; $z \epsilon G'$ on a', b', c' is seen from 5.5.12 to be identical to p(x, z); $x \epsilon F$; $z \epsilon G'$ so these two may be contracted into the single p-function p(x, z); $x \epsilon R$; $z \epsilon G'$. Similarly, the effect of p(x, z); $x \epsilon G$; $z \epsilon H'$ on a', b', c' may be seen from 5.5.12 to be identical to the effect of p(x, z); $x \epsilon F$, $z \epsilon H'$ so these two may be contracted into the single p-function p(x, z); $x \epsilon D$; $z \epsilon H'$. This simplification reduces 5.5.12 to:

5.5.13 Simplified table of the effects of p(x, z); $x \epsilon X$; $z \epsilon Z$ on a', b', c' for all relevant (X, Z).

X	Z	$\Delta_p a'$	$\Delta_p b'$	$\Delta_p c'$
H	R'	0	0	0
R	G'	1	0	-1
H	V or W	0	1	0
D	H'	1	-1	0
G	D'	0	0	0
G	V or W	0	0	1
F	F'	0	0	0
F	V or W	1	0	0

5.6 Effect of the r and a Functions on a Document-Request Pair

The effect of the r-function on a', b', c' is denoted by $\Delta_r a'$, $\Delta_r b'$, $\Delta_r c'$ and the effect of the a-function on a', b', c' is denoted by $\Delta_a a'$, $\Delta_a b'$, $\Delta_a c'$. Suppose that the effect of $t(x, y)$; $x \epsilon X$; $y \epsilon Y$ on a', b', c' is $\Delta_t a'$, $\Delta_t b'$, $\Delta_t c'$ for all terms x belonging to X and all suitable terms y which belong to Y. Suppose also that the effect of $p(x, z)$; $x \epsilon X'$; $z \epsilon Z$ on a', b', c' is $\Delta_p a'$, $\Delta_p b'$, $\Delta_p c'$. From 5.4.4 it is known that $r(x{:}y \rightarrow z) \equiv t(x, y)\ p(x, z)$ where the operations are carried out from left to right. Then the effect of $r(x{:}y \rightarrow z)$; $x \epsilon X''$; $y \epsilon Y$; $z \epsilon Z''$ is given by:

5.6.1
$$\Delta_r a' = \Delta_t a' + \Delta_p a'$$
$$\Delta_r b' = \Delta_t b' + \Delta_p b'$$
$$\Delta_r c' = \Delta_t c' + \Delta_p c'$$

provided that $X'' \subseteq X \wedge X'$ and $Z'' \subseteq Z$.

Also, from 5.4.3 it is known that $a(x \rightarrow z) = p(x, z)$. Thus, the effect of $a(x \rightarrow z)$; $x \epsilon X'$; $z \epsilon Z$ is given by:

5.6.2 $$\Delta_a a' = \Delta_p a'$$

$$\Delta_a b' = \Delta_p b'$$

$$\Delta_a c' = \Delta_p c' \ .$$

Table 5.6.3 below gives a complete list of the single-step r-functions which may be applied to the pseudo-classification. The effects on a', b', c' are deduced by applying 5.6.1 to the table for the t-function given in 5.5.8 and to the table for the p-function given in 5.5.13. In each case the largest set X" contained in both X and X' and the largest set Z" contained in Z are taken. Table 5.6.4 gives a complete list of the single-step a-functions which may be applied to the pseudo-classification. The effects on a', b', c' are given by applying 5.6.2 to the table for the p-function given in 5.5.13.

5.6.3 Table of the Effects of the r-function on a', b', c'.

See Table 5.5.8 t(x,y); x∈X; y∈Y				See Table 5.5.13 p(x,z); x∈X'; z∈Z					r(x:y → z); x∈X"; y∈Y; z∈Z"				
X	$\Delta_t a'$	$\Delta_t b'$	$\Delta_t c'$	X'	Z	$\Delta_p a'$	$\Delta_p b'$	$\Delta_p c'$	X"	Z"	$\Delta_r a'$	$\Delta_r b'$	$\Delta_r c'$
F''_{14}	-1	0	1	R	G'	1	0	-1	F''_{14}	G'	0	0	0
				D	H'	1	-1	0	"	H'	0	-1	1
	ditto			F	F'	0	0	0	"	F'	-1	0	1
				F	V	1	0	0	"	V	0	0	1
				F	W	1	0	0	"	W	0	0	1
F''_{23}	-1	1	0	R	G'	1	0	-1	F''_{23}	G'	0	1	-1
				D	H'	1	-1	0	"	H'	0	0	0
	ditto			F	F'	0	0	0	"	F'	-1	1	0
				F	V	1	0	0	"	V	0	1	0
				F	W	1	0	0	"	W	0	1	0
F''_{31}	-1	0	0	R	G'	1	0	-1	F''_{31}	G'	0	0	-1
				D	H'	1	-1	0	"	H'	0	-1	0
	ditto			F	F'	0	0	0	"	F'	-1	0	0
				F	V	1	0	0	"	V	0	0	0
				F	W	1	0	0	"	W	0	0	0
F''_{41}	0	0	0	R	G'	1	0	-1	F''_{41}	G'	1	0	-1
				D	H'	1	-1	0	"	H'	1	-1	0
	ditto			F	F'	0	0	0	"	F'	0	0	0
				F	V	1	0	0	"	V	1	0	0
				F	W	1	0	0	"	W	1	0	0
H''_1	0	-1	0	H	R'	0	0	0	H''_1	R'	0	-1	0
				R	G'	1	0	-1	"	G'	1	-1	-1
	ditto			H	V	0	1	0	"	V	0	0	0
				H	W	0	1	0	"	W	0	0	0
H''_2	0	0	0	H	R'	0	0	0	H''_2	R'	0	0	0
				R	G'	1	0	-1	"	G'	1	0	-1
	ditto			H	V	0	1	0	"	V	0	1	0
				H	W	0	1	0	"	W	0	1	0
G''_1	0	0	-1	D	H'	1	-1	0	G''_1	H'	1	-1	-1
				G	D'	0	0	0	"	D'	0	0	-1
	ditto			G	V	0	0	1	"	V	0	0	0
				G	W	0	0	1	"	W	0	0	0
G''_2	0	0	0	D	H'	1	-1	0	G''_2	H'	1	-1	0
				G	D'	0	0	0	"	D'	0	0	0
	ditto			G	V	0	0	1	"	V	0	0	1
				G	W	0	0	1	"	W	0	0	1

5.6.4 Table of the effects of the a-function on a', b', c'.

See Table 5.5.13
$a(x \rightarrow z)$; $x \epsilon X$; $z \epsilon Z$

X	Z	$\Delta_a a'$	$\Delta_a b'$	$\Delta_a c'$
H	R'	0	0	0
R	G'	1	0	-1
H	V	0	1	0
H	W	0	1	0
D	H'	1	-1	0
G	D'	0	0	0
G	V	0	0	1
G	W	0	0	1
F	F'	0	0	0
F	V	1	0	0
F	W	1	0	0

The complete algebraic formulation of the perturbations contained in 5.6.3 is carried out as follows. Consider the first perturbation, namely, $r(x{:}y \rightarrow z)$; $x \epsilon F''_{14}$; $y \epsilon Y$; $z \epsilon G'$.

From 5.5.9 $\quad F''_{14} = F''_1 \Lambda F''_4$

from 5.5.4, 6 $\quad = \underline{\text{domain}}\ (t_1;\ R,\ F') \Lambda\ \underline{\text{domain}}\ (t_2;\ D,\ F')$

from 5.4.12 $\quad = (\forall x)\ (\forall y)\ (x \epsilon R \Lambda y;\ x \epsilon D \Lambda y;\ y \epsilon F';\ L_1(y, R);\ L_2(y, D))$.

This provides a description of the terms x and classes y which are allowed as arguments of the perturbation. The complete description is, therefore,

5.6.5 $\quad r(x;\ y \rightarrow z);\ x \epsilon F \wedge y;\ y \epsilon F';\ L_1(y, R);\ L_2(y, D);\ z \epsilon G'$

where from 5.5.2

$$G' = D' \wedge \overline{(R' \wedge D')}$$

$$F' = (R' \wedge D')$$

and from 5.5.2

$$D' = f(d, \underline{C})$$

$$R' = f(R, \underline{C}) .$$

The perturbation is applied to the single triad (x, y, z) which satisfies the condition $x \epsilon F \wedge y;\ y \epsilon F';\ L_1(y, R);\ L_2(y, D);\ z \epsilon G'$. Any (x, y, z) which satisfies this condition will have the effect $\Delta_r a' = 0$, $\Delta_r b' = 0$, $\Delta_r c' = 0$ on a', b', c' as set out in 5.6.3. The choice of (x, y, z) from among the triads which satisfy the condition is made using 4.5.3 and 4.5.4.

5.7 Classification of Perturbations

The perturbations of 5.6.3 are r-type except those for which Z" = V (i.e., z is a new class), which are rn-type. The perturbations of 5.6.4 are c-type except those for which Z" = V, which are cn-type. The types are defined in Section 4.5. There remains, however, the additional grouping into the classes increasing, level, decreasing which are defined in 4.3.3. Consider the general match function of the retrieval model.

From 5.3.4 $\quad l = M(a, b, c, a', b', c')$.

Suppose that $\Delta_r l$ is the increment in l after the application of an r-function. Then,

5.7.1 $$\Delta_r l = M(a, b, c, a' + \Delta_r a', b' + \Delta_r b', c' + \Delta_r c') - M(a, b, c, a', b', c')$$

so that if

5.7.2 $\Delta_r l > 0$ for all a', b', c' then the r-function is <u>increasing</u>

$\Delta_r l = 0$ " <u>level</u>

$\Delta_r l < 0$ " <u>decreasing</u> .

SECTION VI

6. ENUMERATION OF PERTURBATIONS FOR A GIVEN MATCH FUNCTION

6.1 Description of the Match Function

The enumeration of perturbations given in Section 5 was for a general match function which satisfied certain conditions. A match function will now be defined which is a particular case of this general match function. A complete enumeration of the single-step perturbations which affect it will be given. The match function has not been tested in retrieval and has been constructed only with the intention of demonstrating the techniques of Section 5. It is shown, however, that this particular choice of match function obeys the conditions which have been placed upon match functions in Section 3.3.

Suppose that D is a document and R is a request, where R and D are both defined by lists of terms. $\underline{C}$ is the present state of the classification. The match coefficient will be designed to give the extent to which R is included in D, both for terms and for classes. The class-descriptions of R and D are R' and D', respectively, and are defined by:

$$R' = f(R, \underline{C}) \quad \text{and} \quad D' = f(D, \underline{C}) .$$

The purpose of f is to provide a translation of a set of terms X into a set of classes according to the membership of the terms of X to

classes of the classification. The most natural way of doing this is to list all the classes of $\underline{C}$ which contain at least one term in common with X. f may, therefore, be defined by,

6.1.1 $$f(X, \underline{C}) = (C_j | C_{ij} = 1;\ 1;\ T_i \epsilon X) .$$

Terms common both to the document and to the request, however, will necessarily lead to common classes in their class-descriptions. These classes are those which contain at least one of the common terms. If these common terms are removed, the class-descriptions of R and D will have no necessarily included classes. Another definition of f is, therefore,

6.1.2 $$f(X, \underline{C}) = (C_j | C_{ij} = 1;\ T_i \epsilon X \Lambda \overline{(R \Lambda D)})$$

where the T_i are terms of the vocabulary and the C_j are classes of the classification $\underline{C}$. Both of these definitions of f satisfy 5.2.1.

In the comparison of a request and a document it is the included terms and classes which contribute positively to the match coefficient by the Relevance Hypothesis (1.1.2). Suppose that t is the contribution to the match coefficient 1 from the term matches and that c is the contribution from class matches. Suppose further that 1 is defined by:

6.1.3 $$1 = pt + qc$$

where p and q are positive constants. That these constants are positive is a consequence of the Relevance Hypothesis. The contributions t and c to 1 may be designed to give additional emphasis to terms and classes belonging to R but not to D by counting these against the match

coefficient. For terms, this is a slightly ill-advised procedure for it places undue reliance upon standard vocabulary use. It is a problem more properly dealt with using the term classification. A simple definition of t is,

6.1.4 $$t = N(R \wedge D)/N(R) .$$

For classes, however, the position is slightly different. The classes of the classification C are intended to represent concepts which are apposite to the document collection. If a document is relevant to a request, it is, therefore, expected that the class-descriptions conform to a higher degree than the term descriptions. It may be argued, therefore, that the absence of a complete concept from a document detracts more from the relevance of the document to the request than an absent term. A plausible expression for c which will serve for this example is,

6.1.5 $$c = (N(R' \wedge D')/N(R')) - ((NR' \wedge \overline{(R' \wedge D')})/N(R')) .$$

Thus, if there are proportionately more classes of the request contained in the document than not, then the contribution to the match coefficient is positive. Otherwise, it is negative.

The role of p and q can now be interpreted in terms of the class and term matches. The values of p and q may be used to increase the importance of class matches compared with term matches, or vice versa. In this example it will be assumed that in the evaluation of the match coefficient a class match is equivalent to a term match. Accordingly, in the notation of 5.3.2, 6.1.4, and 6.1.5 become:

6.1.6 $t = a/n$

6.1.7 $c = (a' - b')/n'$

whence from 6.1.3:

6.1.8 $l = a/n + (a' - b')/n'$.

This, together with 6.1.2, is to be regarded as the match function M to be used throughout this example.

It will now be shown that this satisfies the conditions which have been imposed upon the behaviour of match functions. 6.1.8 satisfies 3.3.2 since it is a monotonically increasing function of a and a' for a specific document and a specific request. It also satisfies 3.3.3 because it is independent of N', 3.3.5 trivially and 3.3.6 since it does not depend on N. Provided that the request has at least one term, l is finite. In addition, since $0 \leq t \leq 1$ and $-1 \leq c \leq 1$, the match coefficient is bounded above and below. It reaches its lower and upper bound if the term and class descriptions of the request and document are mutually disjoint; and reaches its upper bound if the term and class descriptions of the request are entirely contained in those of the document. 6.1.8, therefore, satisfies 3.3.4, and, therefore, all the conditions of Section 3.3. The upper and lower bounds of l are given, respectively, by,

$$l^+ = 2 \quad \text{and} \quad l^- = -1 \ .$$

The critical value of the match coefficient is, therefore, given, from 3.3.8, by:

$$l_0 = \tfrac{1}{2} .$$

It is important to recollect that none of the changes caused by perturbations may alter the term descriptions of R and D. Therefore, for a particular document and request, t will remain constant. All variability in l derives from c. From 6.1.7:

6.1.9 $\quad c = 1 - 2b'/n'$

since $a' + b' = n'$. Therefore, it may be seen that the match function 6.1.8 which involves decreasing the match coefficient with the number of classes in the request but not in the document, may be regarded as equivalent to a match function in which the proportion of shared terms and shared classes is measured, and in which matches on classes count twice as much as matches on terms. This equivalent match function has $p = 1$ and $q = 2$, the constant 1 being ignored.

6.2 Enumeration of Perturbations

Let $\Delta_r l$ be the change in the match coefficient due to an r-function. Then form 6.1.3:

$$\Delta_r l = \Delta_r t + \Delta_r c$$

$$= \Delta_r c \quad (\Delta_r t = 0 \text{ since the term description of D and R remain unchanged})$$

from 6.1.9

$$\Delta_r 1 = -2 \cdot \left(\frac{b' + \Delta_r b'}{n' + \Delta_r n'} - \frac{b'}{n'}\right)$$

$$= -2 \cdot \left(\frac{n' \Delta_r b' - b' \Delta_r n'}{n'(n' + \Delta_r n')}\right)$$

$$= 2 \cdot g(\Delta_r b', \Delta_r n', b', n') \text{ say,}$$

where $n' = a' + b'$ and $\Delta_r n' = \Delta_r a' + \Delta_r b'$. The values of $\Delta_r b'$ and $\Delta_r n'$ may be determined for each r-function from 5.6.3 and for each a-function form 5.6.4. The pairs of values of these increments which occur for the present match function are (1, 1), (1, 0), (0, -1), (0, 0), (-1, -1), (-1, 0), (0, 1). The increases in 1 corresponding to these changes are given in 6.2.1 below, together with the grouping of the perturbations which produce these changes into <u>increasing</u>, <u>level</u>, and <u>decreasing</u>.

6.2.1 $g(1,1,b',n') = \frac{n' - b'}{n'(n' + 1)} > 0$ since $n' > b'$. <u>Decreasing</u>

$g(1,0,b',n') = \frac{1}{n' + 1} > 0$ "

$g(0,-1,b',n') = \frac{b'}{n'(n' - 1)} > 0$ "

$g(0,0,b',n') = 0$ <u>Level</u>

$g(-1,-1,b',n') = \frac{-n' + b'}{n'(n' - 1)} < 0$ since $n' > b'$ <u>Increasing</u>

$$g(-1,0,b',n') = \frac{-1}{n'} < 0 \qquad \underline{\text{Increasing}}$$

$$g(0,1,b',n') = \frac{-b'}{n'(n'+1)} < 0 \quad . \qquad "$$

For a particular match function, a number of perturbations which, although distinct according to the general theory of Section 5, may be expressible as a single perturbation with suitably adjusted conditions on the arguments. The conflation of perturbations in this way is called simplification. Suppose that $u_1, \ldots, u_k$ where $k \leq 3$ is a subset of a', b', c'. Then from 5.7.1:

$$\Delta_r 1 = M(a,b,c,u_1 + \Delta_r u_1, \ldots, u_k + \Delta_r u_k) - M(a,b,c,u_1,\ldots,u_k) .$$

Suppose that $r(x:y \rightarrow z)$; $x \epsilon X_1$; $y \epsilon Y_1$; $z \epsilon Z_1$ and $r(x:y \rightarrow z)$; $x \epsilon X_2$; $y \epsilon Y_2$; $z \epsilon Z_2$ are two perturbations and suppose that their effects on $u_1,\ldots,u_k$ are $\Delta_r u'_i$ and $\Delta_r u''_i$, respectively. Then the two perturbations are said to be <u>equivalent</u> if:

6.2.2 $\quad \Delta_r u'_i = \Delta_r u''_i$ for all $i \leq k$.

Thus, if two perturbations are equivalent, then they change the match coefficient by the same amount. Suppose that the two perturbations are equivalent and suppose further that X_1 and X_2 are identical sets and that Y_1 and Y_2 are identical sets. Then the two perturbations may be simplified to $r(x:y \rightarrow z)$; $x \epsilon X_1$; $y \epsilon Y_1$; $z \epsilon Z_1 \vee Z_2$ provided that Z_1, Z_2 belong to R' or D'. A distinction is maintained between 'new classes',

classes belonging to R' or D', and classes belonging to the classification but not to R' or D' (see 5.5.10). Suppose on the other hand that Z_1 and Z_2 are the same set. Then, providing certain conditions are satisfied by X_1, X_2, X_3, Y_1, Y_2, Y_3 the two perturbations may be simplified to $r(x:y \rightarrow z)$; $x \epsilon X_3$; $y \epsilon Y_3$; $z \epsilon Z_1$. A general discussion of simplification is not embarked upon here since it will be seen from 6.2.3 that only two simplifications may be carried out. Analogous statements hold for the a-function.

In 6.2.3 and 6.2.4 the perturbations for the match function defined in 6.1.8 are given together with their type (c, r, cn, rn) and their effect on the match coefficient (<u>increasing</u> (+), <u>level</u> (0), <u>decreasing</u> (-)). In 6.2.3 two simplifications have been carried out. One involves perturbations for which X" is either F''_{14} or F''_{31} and the other involves perturbations for which X" is either G''_1 or G''_2. The complete list of perturbations, both r-functions and a-functions, is written out in full in 6.2.5. The perturbations are grouped according to increasing, level, and decreasing and each of these groups is further divided into c, r, cn, rn type perturbations.

6.2.3 Table of r-functions for $l = (a/n) + (a' - b')/n'$

$r(x:y \rightarrow z)$; $x \epsilon X''$; $y \epsilon Y$; $z \epsilon Z''$

X''	Z''	Type	$\Delta_r b'$	$\Delta_r n'$	Effect on l
F''_1	G'	r	0	0	0
"	H'	r	-1	-1	+
"	F'	r	0	-1	-
"	V	r	0	0	0
"	W	rn	0	0	0
F''_{23}	G'	r	1	1	-
"	H'	r	0	0	0
"	F'	r	1	0	-
"	V	r	1	1	-
"	W	rn	1	1	-
F''_{42}	G'	r	0	1	+
"	H'	r	-1	0	+
"	F'	r	0	0	0
"	V	r	0	1	+
"	W	rn	0	1	+
H''_1	R'	r	1	1	-
"	G'	r	-1	0	+
"	V	r	0	0	0
"	W	rn	0	0	0
H''_2	R'	r	0	0	0
"	G'	r	0	1	+
"	V	r	1	1	-
"	W	rn	1	1	-
D	H'	r	-1	0	+
"	D'	r	0	0	0
"	V	r	0	0	0
"	W	rn	0	0	0

6.2.4 Table of a-functions for $l = (a/n) + (a' - b')/n'$

$a(x \rightarrow z)$; $x \in X$; $z \in Z$

X	Z	Type	$\Delta_a b'$	$\Delta_a n'$	Effect on l
H	R'	c	0	0	0
R	G'	c	0	1	+
H	V	c	1	1	-
H	W	cn	1	1	-
D	H'	c	-1	0	+
G	D'	c	0	0	0
G	V	c	0	0	0
G	W	cn	0	0	0
F	F'	c	0	0	0
F	V	c	0	1	+
F	W	cn	0	1	+

6.2.5 Classified list of perturbations for $l = (a/n) + (a' - b')/n'$

Decreasing

c-type

$a(x \rightarrow z)$; $x \epsilon H$; $z \epsilon V$

r-type

$r(x{:}y \rightarrow z)$; $x \epsilon R \wedge y$; $y \epsilon F'$; $L_1(y,R)$; $z \epsilon F'$

$r(x{:}y \rightarrow z)$; $x \epsilon F \wedge y$; $y \epsilon F'$; $L_2(y,R)$; $L_1(y,D)$; $z \epsilon G'$

$r(x{:}y \rightarrow z)$; $x \epsilon F \wedge y$; $y \epsilon F'$; $L_2(y,R)$; $L_1(y,D)$; $z \epsilon F'$

$r(x{:}y \rightarrow z)$; $x \epsilon F \wedge y$; $y \epsilon F'$; $L_2(y,R)$; $L_1(y,D)$; $z \epsilon V$

$r(x{:}y \rightarrow z)$; $x \epsilon R \wedge y$; $y \epsilon H'$; $L_1(y,R)$; $z \epsilon R'$

$r(x{:}y \rightarrow z)$; $x \epsilon R \wedge y$; $y \epsilon H'$; $L_2(y,R)$; $z \epsilon V$

cn-type

$a(x \rightarrow z)$; $x \epsilon H$; $z \epsilon W$

rn-type

$r(x{:}y \rightarrow z)$; $x \epsilon F \wedge y$; $y \epsilon F'$; $L_2(y,R)$; $L_1(y,D)$; $z \epsilon W$

$r(x{:}y \rightarrow z)$; $x \epsilon R \wedge y$; $y \epsilon H'$; $L_2(y,R)$; $z \epsilon W$

Level

c-type

$a(x \rightarrow z);\ x \epsilon H;\ z \epsilon R'$

$a(x \rightarrow z);\ x \epsilon G;\ z \epsilon D'$

$a(x \rightarrow z);\ x \epsilon G;\ z \epsilon V$

$a(x \rightarrow z);\ x \epsilon F;\ z \epsilon F'$

r-type

$r(x{:}y \rightarrow z);\ x \epsilon R \wedge y;\ y \epsilon F';\ L_1(y,R);\ z \epsilon G'$

$r(x{:}y \rightarrow z);\ x \epsilon R \wedge y;\ y \epsilon F';\ L_1(y,R);\ z \epsilon V$

$r(x{:}y \rightarrow z);\ x \epsilon F \wedge y;\ y \epsilon F';\ L_2(y,R);\ L_1(y,D);\ z \epsilon H'$

$r(x{:}y \rightarrow z);\ x \epsilon F \wedge y;\ y \epsilon F';\ L_2(y,R);\ L_2(y,D);\ z \epsilon F'$

$r(x{:}y \rightarrow z);\ x \epsilon R \wedge y;\ y \epsilon H';\ L_1(y,R);\ z \epsilon V$

$r(x{:}y \rightarrow z);\ x \epsilon R \wedge y;\ y \epsilon H';\ L_2(y,R);\ z \epsilon R'$

$r(x{:}y \rightarrow z);\ x \epsilon D \wedge y;\ y \epsilon G';\ z \epsilon D'$

$r(x{:}y \rightarrow z);\ x \epsilon D \wedge y;\ y \epsilon G';\ z \epsilon V$

cn-type

$a(x \rightarrow z);\ x \epsilon G;\ z \epsilon W$

rn-type

$r(x{:}y \rightarrow z);\ x \epsilon R \wedge y;\ y \epsilon F';\ L_1(y,R);\ z \epsilon W$

$r(x{:}y \rightarrow z);\ x \epsilon R \wedge y;\ y \epsilon H';\ L_1(y,R);\ z \epsilon W$

$r(x{:}y \rightarrow z);\ x \epsilon D \wedge y;\ y \epsilon G';\ z \epsilon W$

Increasing

c-type

$a(x \rightarrow z);\ x \epsilon R;\ z \epsilon G'$

$a(x \rightarrow z);\ x \epsilon D;\ z \epsilon H'$

$a(x \rightarrow z);\ x \epsilon F;\ z \epsilon V$

r-type

$r(x{:}y \rightarrow z);\ x \epsilon R \wedge y;\ y \epsilon F';\ L_1(y,R);\ z \epsilon H'$

$r(x{:}y \rightarrow z);\ x \epsilon F \wedge y;\ y \epsilon F';\ L_2(y,R);\ L_2(y,D);\ z \epsilon G'$

$r(x{:}y \rightarrow z);\ x \epsilon F \wedge y;\ y \epsilon F';\ L_2(y,R);\ L_2(y,D);\ z \epsilon H'$

$r(x{:}y \rightarrow z);\ x \epsilon F \wedge y;\ y \epsilon F';\ L_2(y,R);\ L_2(y,D);\ z \epsilon V$

$r(x{:}y \rightarrow z);\ x \epsilon F \wedge y;\ y \epsilon H';\ L_1(y,R);\ z \epsilon G'$

$r(x{:}y \rightarrow z);\ x \epsilon F \wedge y;\ y \epsilon H';\ L_2(y,R);\ z \epsilon G'$

$r(x{:}y \rightarrow z);\ x \epsilon D \wedge y;\ y \epsilon G';\ z \epsilon H'$

cn-type

$a(x \rightarrow z);\ x \epsilon F;\ z \epsilon W$

rn-type

$r(x{:}y \rightarrow z);\ x \epsilon F \wedge y;\ y \epsilon F';\ L_2(y,R);\ L_2(y,D);\ z \epsilon W$

SECTION VII

7. IMPLEMENTATION

The most suitable programming language for implementing the perturbations developed by this paper seems to be provided by the STDS System (Set-theoretic data structures) of Childs (11).

REFERENCES

1. Salton, G., Computer Evaluation of Indexing and Text Processing. J. ACM., 15(1), 1968, pp. 8-36.

2. Sparck Jones, K. and Jackson, D. M., Current Approaches to Classification and Clump-Finding at the CLRU. Computer Journal, 10, 1967, pp. 29-37.

3. Doyle, L. B., Is Automatic Classification a Reasonable Application of Statistical Analysis of Text? ACM., 12, 1965, pp. 473-489.

4. Sparck Jones, K. and Jackson, D. M., The Use of the Theory of Clumps for Information Retrieval. Report on the OSTI-supported project at the Cambridge Language Research Unit, ML 200, 1967. (mimeo)

5. Salton, G. and Lesk, M., The SMART Automatic Document Retrieval System-An Illustration. Comm. ACM., 8(6), 1965, pp. 391-398.

6. Swets, J. A., Effectiveness of Information Retrieval Methods. Bolt Beranek and Newman, Rept. 1499, Cambridge, Mass., April, 1967.

7. Cleverdon, C. and Keen, M., Factors Determining the Performance of Indexing Systems, Vols. 1 and 2, ASLIB, Cranfield Research Project, 1966.

8. Resnick, A. and Savage, T. R., The Consistency of Human Judgments of Relevance. American Documentation, 15(2), 1964, pp. 93-95.

9. Maron, M. E. and Kuhns, J. L., On Relevance, Probabilistic Indexing and Information Retrieval. J. ACM., 7, 1960, pp. 216-244.

10. Jackson, D. M., A Note on a Set of Functions for Information Retrieval. Information Storage and Retrieval, 1969 (in press).

11. Childs, D. L., Description of a Set-Theoretic Data Structure. Fall Joint Computer Conference, Paper Number 82, 1968, pp. 557-564.