A LEARNING ALGORITHM APPLIED TO DOCUMENT REDESCRIPTION

Michael D. Gordon
Computer and Information Systems
Graduate School of Business Administration
The University of Michigan
Ann Arbor, MI 48109-1234

1. Introduction

Relevance feedback in document retrieval usually involves an inquirer revising a query to obtain better retrieval effectiveness [Rocchio] and [Salton]. The revised query is adjusted to correspond to the descriptions of known documents which the inquirer has found relevant. Several problems exist with such methods, however, and will be described in this paper.

Less frequently, the descriptions of documents, themselves, have been altered on the basis of inquirer feedback. The advantages that arise from allowing a document description to change over time will be discussed. The heart of this paper will then be devoted to discussing how an algorithm used in artificial intelligence can be used to help redescribe documents. A simulation of a document retrieval system subject to such redescription was conducted, and the results of the simulation are described.

2. Query Modification

Document retrieval is known to be a trial and error process [Swanson]. As a result, it is understandable that attempts have been made to build feedback into the process to improve retrieval effectiveness. Document retrieval involves each of the following: inquirers identifying their information need with machine processable queries; the subject contents of documents being represented and stored in a bibliographic database; and a matching function which the retrieval system has been programmed to use to select documents in response to a given query.

It would be possible, then, to try to modify the workings of a document retrieval system in various ways (each in an attempt to improve the effectiveness of the system): by reformulating the inquirer's query; by changing the descriptions of the documents stored in the database; by adjusting the matching function; or even by making several of these changes at once.

Salton and Rocchio have studied automatic query reformulation (or relevance feedback) and have been able to improve retrieval effectiveness in controlled, experimental retrieval environments [Salton] [Rocchio]. In essence, the various techniques they offer for query reformulation take the descriptions of documents known to satisfy the inquirer's need and automatically build new queries out of them. Oddy puts the query adjustment back in the searchers' hands (although, strictly, a searcher is not making a query). In his system, the user learns more about the structure of the document database and, hopefully, becomes more able to indicate precisely what kinds of documents he or she is looking for [Oddy].

A problem with such query reformulation systems is that documents may be poorly or inconsistently described. There is known variability in the way trained indexers describe the same documents [Zunde and Dexter], so there is really no reason to be optimistic that a given document is described as well as possible. (Similarly, automatic indexing carries no guarantees that the statistical occurrences of tokens in text can be used to adequately describe that document.) Worse, documents which satisfy the need of a given inquirer may be described quite differently. Any attempt then to alter a query to be more like the description of one of these relevant documents may bring about worse matching relative to another. Thus, relevance feedback can succeed to the degree that there are clearly identifiable, "tight" document clusters [Salton and McGill].

3. Document Redescription

In contrast to redescribing queries, the descriptions of documents, themselves, may be redescribed. Simplistically, if a retrieval system were devoted to one inquirer, document descriptions could be changed in response to his or her relevance assessments. The relevant document which the system has predicted is only marginally relevant to the inquirer's query would be redescribed, if discovered, to increase the likelihood of its being furnished the next time the inquirer makes a similar query. (Exact matching retrieval functions can modify documents somewhat similarly.)

In multi-user retrieval systems, the same sort of redescription is possible. Instead of modifying a document's description

to match better the appropriate query (or queries) of a single inquirer, a document may be redescribed to match better the "relevant queries" of any inquirer who finds that document satisfies his or her information need. Then, the next time any of these inquirers searches for that document with its modified description, chances are he or she should find it with less effort (fewer searches) or with a higher prediction of probabilistic relevance by the retrieval system. To the extent that other inquirers make similar queries, they, too, will find the system predicting relevance more effectively.

A question which naturally arises is: why is it better to redescribe the descriptions of documents than the queries put forth to retrieve them? In part, the question is wrong. To the extent possible, the two types of modifications should be used together since they are not incompatible. In fact, it is the communication back and forth between inquirer and retrieval system that should be stressed as a principle of document retrieval [Gordon], [Oddy].

From another vantage, though, document redescription can be regarded as more natural and more effective than query reformulation. Document retrieval systems which ultimately hope to provide a natural language interface can make use of document redescription so that a document's descriptions and the queries made to retrieve it become more alike. Documents become described similarly to the way they are naturally looked for instead of queries being reformulated to match document descriptions which may be rather arbitrarily constructed. Additionally, as we have seen, the inaccuracy of a document description and the dissimilar descriptions of documents which are relevant to the same query may militate against effective query reformulation. In redescribing documents, the "inaccuracies" or "dissimilarities" that must be accounted for arise solely from the way people naturally use language (and issue queries to express their information needs). These variations in linguistic style are built into the problem of document retrieval. The better able we are in accommodating this variation, the better our systems will be. In other words, querying (using language) is (nearly) a natural phenomenon, but document descriptions are the artifacts of indexing. Both are attempts to "name" the same thing. So it seems sensical to try to modify the "artificial" name (index) to resemble the natural.

A previous attempt at adjusting term weights is reported by Brauen [Brauen]. Despite its promise, this approach performed worse when it received feedback information concerning both successful and unsuccessful searches than when it only received the former. Also, "control" queries, which should not necessarily have been more easily retrieved after document modification, exhibited even better recall-precision performance as a result of adjusting term weights than did the "test" queries toward which document redescription was directed.

4. Genetic Algorithm

An adaptive algorithm, which has application in many fields, can be used to improve the description of document descriptions [Holland 1975]. This adaptive algorithm (or "genetic algorithm," since it is based on a genetic metaphor) is currently being used in artificial intelligence research to help promote learning [Holland 1983]. As I have pointed out, retrieval systems can learn about the the relevance relationships between user queries and documents. Thus, the genetic algorithm can be seen as an effective tool for document redescription.

The genetic algorithm operates essentially as follows:

Repeat

1) Measure the performance (worth) of each competing object in a (fixed-size) set.
2) Replace the set of objects:
   First, throw away the current set of objects.
   Then build new objects out of old object parts, using more parts of the objects with higher worth. Each of these new objects will likely be different than all objects in the previously discarded set.

Until some criterion is attained.[1]

The algorithm attempts to mimic genetics, promoting a population built up of parts ("genes") of its fittest members. As in genetics, succeeding generations introduce variety (new "objects", i.e. people, who resemble their forebears but are not identical to them). A simplifying, non-biological example may provide a better feeling for the algorithm. Suppose the performance (worth) of a car can be measured as a function of its speed and the comfort it provides its driver. Then the a Porsche may have great worth (because it's fast) while a Rolls Royce may have great worth (because of its comfort). The genetic algorithm might "build" a new car (Porsche Royce) out the parts of both cars, trying to attain speed and comfort at once.

In genetics, a gene carried by the fittest members of the population will proliferate from one generation to the next (assuming these fit members have disproportionately more offsprings than their less fit counterparts). In this way, succeeding generations tend to be fitter than their parents. The genetic algorithm operates by determining the fittest members in a given generation, ensuring that they are disproportionately represented the next generation, and introducing variety in the population to try to "build" offspring fitter than their parents. Further, the algorithm selects not only good "genes" but good "gene combinations." That is, even if many traits ("genes") of an object interact in complex ways, the genetic algorithm works to find these viable combinations.

[1] See appendix A for a more detailed discussion of the algorithm. This appendix is best read following section 6.

5. Adaptive Document Redescription

Document description may be cast as a problem of "building" adequate representations of documents. The "genes" of a document's description are the terms used to describe it (or, alternatively, the weights of those terms, as the case may be). What genetic adaptation can do, then, is to "build" fitter and fitter document descriptions in succeeding generations. The more the system learns about the requests used to retrieve a given document, the better it will become subsequently in furnishing that document to "relevant" queries.

6. Simulation of Adaptive Document Description

A document retrieval simulation was conducted to determine the effectiveness of using the genetic algorithm to redescribe documents. Eighteen documents had their subject descriptions altered. The way one document was redescribed was unaffected by the redescription of any other document. What follows is a brief description of the redescription of one of these eighteen documents:

First, since the genetic algorithm conducts a "competition" among competing objects in some system, a given document was initially described by approximately seventeen undergraduates who had just read it. Each of these students chose from a closed list of subject phrases those phrases he or she felt "definitely" (or almost definitely) described that document. Thus, one student might have described a document as being about the four subject phrases: office automation, improving productivity, office of the future, personal work stations. Another student may have described the document with some of these terms, too, or any of the (approximately thirty) terms provided to choose from. In this way, the document was described separately (and independently) by approximately seventeen individuals. (The simulation experiment converted these descriptions into seventeen binary strings--or vectors--for use by the genetic algorithm.)

The action of the simulation was to take a set of queries, knowing in advance that the given document was relevant to each of them, and determine how good each of the competing document descriptions was at matching this query set. This determination made, the genetic algorithm would intervene, replacing the given set of seventeen descriptions of the document with seventeen new ones. The adequacy of each description in this new set was measured relative to the same set of seventeen original queries, and again the set of descriptions was replaced by a new set of seventeen document descriptions created by the genetic algorithm. Each round of evaluation-replacement constituted one generation, and the simulation was run for forty generations. (See Figure 1.) At the end of the fortieth generation, the effectiveness of the prevailing set of descriptions was compared to that of the original set of descriptions. An explanation of the method of computing the effectiveness of a document description (or set of descriptions) with respect to the fixed set of of "relevant queries" follows.

No document description can be perfect (in the sense that it filters exactly those queries to which it is relevant from the others) because of the variance among inquirers looking for the same (or similar) documents. But, the better the overlap (agreement) between a document's subject description and a "relevant query," the better we might feel the description is. Similarly, we may measure how well a given description of a document overlaps each of a set of "relevant queries" to tell how well it does at matching this set as a whole. (The term "relevant query" is simply shorthand for "a query to which a given document is known to be relevant." "Relevant query set" is similarly defined.)

Accordingly, a Jaccard's score was used in evaluating a document description relative to a query. Considering X to be the set of subject phrases used in a query and Y to be the set of phrases used in a (single) description of a document relevant to that query, Jaccard(X,Y) = #(X intersect Y) / #(X union Y), # being the cardinality of the set. Relative to a given query, two competing descriptions of the same relevant document can be compared by Jaccard's scores: the higher the Jaccard's score the better the description. Similarly, relative to a set of "relevant queries," the adequacy of a document description may be judged by determining its average Jaccard's score with respect to that set. In fact, the genetic algorithm operated in exactly this way: determine which descriptions in the current set have the highest average Jaccard's scores relative to the (fixed) set of relevant queries, and create new descriptions the next generation out of the parts of those descriptions exhibiting the highest average Jaccard's matching scores this generation. (See Figure 2.)

The same technique of making Jaccard's score comparisons was used to determine the effectiveness of adaptation. That is, each "generation" is characterized by the seventeen descriptions it uses to describe a given document. And, for each generation, there is an "overall" average Jaccard's matching score relating the set of "relevant queries" (fixed from generation to generation) to the current set of descriptions. The higher this "overall" average matching score during a given generation, the more similar the prevailing set of document descriptions is to the set of relevant queries. (See again Figure 2.) Thus, the goal of adaptation was, first of all, to produce superior "overall" average matching in generation 40 (at the end of the simulation) than in generation 1 (with the original set of descriptions). Indeed, for each of the eighteen documents redescribed, there was an improvement in "overall" average matching from generation 1 to 40 (averaging 24% improvement in Jaccard's score; see Figure 3).

Since each of the eighteen documents showed better "overall" average matching after adaptation, the description of any given document had thus been changed so that it would now be easier to retrieve, on average, by any one of the queries to which it was

relevant. (Easier, because it matched better these queries.) Additionally, a control used in this phase of the simulation revealed that redescribing documents with a genetic algorithm could be achieved without an accompanying degradation in fallout. That is, after adaptation, a given document would be more easily retrieved by a "relevant query," but not simply by any query.

This control, like the simulation, considered each document individually. For any document, the following question was answered: Will a set of queries that the document is not relevant to--even though the document's description bears considerable resemblance to these queries--be more likely to retrieve that document after the genetic algorithm redescribes the document? That is: will redescribing a document to improve its retrievability by "relevant queries" be offset by its improved retrievability also by "non-relevant queries?"

The same questionnaire used by undergraduates to supply initial document descriptions contained enough information to create a set of "non-relevant queries." Simply, all subject phrases that the reader of the document said were "somewhat" (or a little less) about the document were taken to represent a query to which the document was not relevant. For instance, in saying a given document was "somewhat" about "artificial intelligence," "medicine," and "expert systems," the contention was that, if the describer were asking for documents concerning those terms, that document would not be considered relevant. By having (approximately) seventeen people make such evaluations about a given document, seventeen "non-relevant queries" were obtained for each document.

The question that needed answering, therefore, was: what happens to "overall" matching between this document and its non-relevant queries after the document is redescribed to match better its "relevant queries?" In evaluating this question separately for each of the eighteen documents to which it applied, the answer became: as documents are redescribed, there usually is some (undesired) increase in "overall" average Jaccard's score relative to non-relevant queries, too. However, and importantly, the redescription strongly favored the relevant queries. Specifically, there was nearly five times the improvement in Jaccard's score matching for "relevant queries" than for "non-relevant queries" as the result of adaptation. Just as important, for seventeen of the eighteen documents, the "desired increase in matching" (between relevant queries and redescribed documents) exceeded the "undesired increase" (between non-relevant queries and redescribed documents. (See Table 1.) In terms of real life retrieval, this suggests that deciding which documents are likely to be relevant to a given query is an easier task because there is a wider gap between the Jaccard's scores calculated for relevant and non-relevant documents.

In a second, and more ambitious, phase of the simulation, the attempt was made to alter a document's description so that the document would be more retrievable by relevant queries but less retrievable by non-relevant queries. Documents were initially given multiple descriptions just as they were in the first phase of the simulation: by taking, for each of approximately seventeen undergraduates, those subject phrases the student thought "definitely" (or almost definitely) described the document's subject content. The goal of the adaptation was to alter this set of descriptions so that their "overall" average matching score relative to relevant queries would rise but their "overall" average matching score relative to "non-relevant queries" would fall. (The procedure for obtaining both relevant and non-relevant queries has been explained.) That is, try to redescribe documents so that they are more likely to be retrieved when they should be and less likely when not. (See Figure 4.)

In this spirit, forty generations of document redescription took place (independently for each of eighteen documents, each document having its own initial descriptions and relevant and non-relevant query sets). At the end of forty generations of adaptation, each

generation attempting to build new descriptions out of the best parts of previously evaluated descriptions, two determinations were made: One, did the final set of descriptions associated with a document match better the relevant queries than the descriptions originally assigned to the document? Two, did the (same) final set of descriptions show less resemblance (match worse) the non-relevant queries than the original descriptions?

The results of simulating the redescription of eighteen documents indicated that both of these effects were achieved. For each of the eighteen documents, redescription improved the "overall" average matching score relative to the relevant queries (by an average of 19.09%). Additionally, the same redescription brought about less resemblance between document descriptions and non-relevant queries in fifteen cases out of eighteen (producing an average worsening in resemblance of 24.81% across all documents). (See Figure 5 and Table 2.) In sum, then, the genetic algorithm was used to build new descriptions of documents that come closer to doing their job: distinguishing the inquiries of those who are interested in a document from those who are not.

7. Summary

The basic argument advanced in this paper has been that redescribing the subject content of documents in light of inquirers' relevance assessments ought to be conducted. The genetic algorithm, a probabilistic algorithm having application to learning and artificial intelligence, has been shown to be effective in governing redescription.

Document redescription governed by adaptation has shown to be successful in both of these ways: by making document descriptions match better "relevant queries"; and by making document descriptions match less well "non-relevant queries" at the same time.

In real world retrieval, the same inquirers do not reappear again and again, always making identical queries (as the simula-

tion suggests they do). But, past inquiring behavior is the best evidence we have for predicting future inquiring behavior. Further, the genetic algorithm would be sensitive to changes in the nature of the "relevant queries" issued for some document.

Inquirers who attempt to name those subjects about which they are interested in retrieving documents are making many fine distinctions. In fact, the questionnaire data obtained in this research showed that, all documents and all terms considered, over nine times in ten, at least one respondent felt a given subject term applied to a document while other respondents felt it did not, or vice versa. In a "descriptively rich" retrieval environment like that in this study, there may be twenty or more plausible descriptions with which a document can be described and, consequently, which an inquirer may choose in searching for the document. With so much variation in the terms inquirers employ, the document redescription problem is quite difficult, and is best handled by a powerful algorithm (such as the genetic algorithm used in this study). In fact, a deterministic algorithm which simply built document descriptions term by term according to the consensus of relevant queries attained Jaccard's matching scores which the genetic algorithm was able to improve upon by 25%. Further, genetic adaptation makes no assumptions about statistical term independence, instead seeking the best combinations of subject terms and building descriptions from them.

In all, the document description process ought to be regarded as dynamic, instead of being done once and never again. In this way, we can improve the odds that the "right inquirers" will find the "right documents," and the wrong inquirers will have fewer irrelevant documents to look through. The genetic algorithm has proven to be a successful redescription technique in the present simulation.

<desc_x_11>	<desc_x_21>	...	<desc_x_M1>
<desc_x_12>	<desc_x_22>	...	<desc_x_M2>
.	.		.
.	.		.
<desc_x_1N>	<desc_x_2N>	...	<desc_x_MN>

======= Adaptation =====>

Generation 1	Generation 2	Generation 40
Descrip set	Descrip set	Descrip set

This figure shows the set of descriptions that describes document-x at various stages of the simulation. In each generation, document_x is represented by a set of N descriptions. Each description is a set of subject terms. Desc_x_ij is the j-th description of document-x in generation-i. Adaptation occurs and the set of descriptions in one generation is replaced by the set of descriptions in the next. N averaged approximately 17, all documents considered.

Figure 1--Evolution of document descriptions

	relev_x_q1	...	relev_x_qM	Avg Matching Score
desc_x_g1	J(g1,q1)	...	J(g1,qM)	$1/M \Sigma J(g1,qi)$
.	.		.	.
.	.		.	.
.	.		.	.
desc_x_gN	J(gN,q1)	...	J(gN,qM)	$1/M \Sigma J(gN,qi)$

N descriptions of document-x in generation-g

Overall average, $G_g = \frac{1}{M * N} \Sigma_k \Sigma_i J(gk,qi)$

Each of document-x's M relevant queries is matched against each of the document descriptions in force in generation-g. The Jaccard's match between relevant query relev_x_qi and document description desc_x_gj is indicated by J(gj,qi). Row averages give "average matching scores" for each document description. G_g, the grand average, gives the "overall average matching score" for the document descriptions in force in the current generation, g.

A set of descriptions of document-x which produces an overall averge matching score greater than G_g relative to the same relevant queries is an improvement on the generation-g set of descriptions.

Figure 2--Matching of descriptions and relevant queries

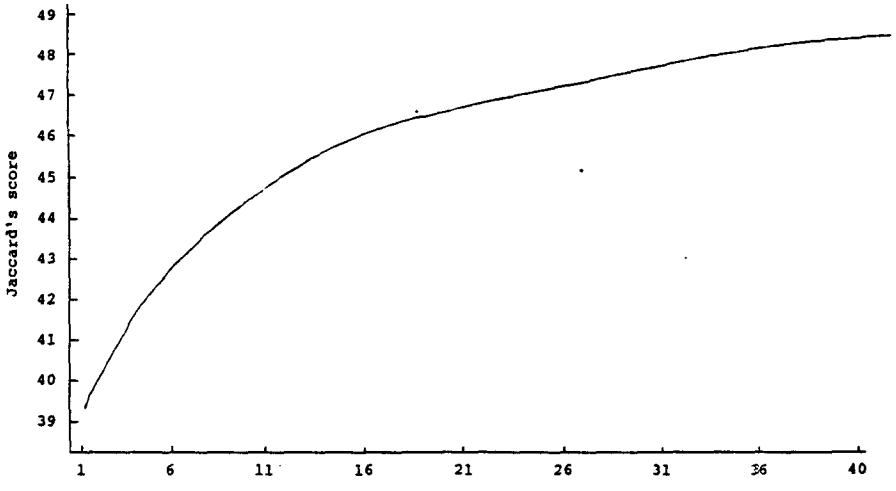

Figure 3--Jaccard's score improvement--all documents combined

	relev_x_q1	...	relev_x_qM	Avg Recall Matching Score
desc_x_g1	J(g1,q1)	...	J(g1,qM)	$1/M \sum_i J(g1,qi)$
.	.		.	.
.	.		.	.
.	.		.	.
desc_x_gN	J(gN,q1)	...	J(gN,qM)	$1/M \sum_i J(gN,qi)$

N descriptions of document-x in generation-g

Grand average, G_g, = $\frac{1}{M * N} \sum_k \sum_i J(gk,qi)$

****

	non-rel_x_q1	...	non-rel_x_qM	Avg Fallout Matching Score
desc_x_g1	J(g1,q1)	...	J(g1,qM)	$1/M \sum_i J(g1,qi)$
.	.		.	.
.	.		.	.
.	.		.	.
desc_x_gN	J(gN,q1)	...	J(gN,qM)	$1/M \sum_i J(gN,qi)$

N descriptions of document-x in generation-g

Grand average, G'_g, = $\frac{1}{M * N} \sum_k \sum_i J(gk,qi)$

Each document description is matched with each relevant query and also with each non-relevant query. For each document description, a matching score is calculated with respect to the relevant query set (row averages above the starred line), and also an average matching score is calculated with respect to the non-relevant query set (row averages below the starred line).

Note that, above the dotted line, J(gi,qj) indicates the Jaccard match between description desc_x_gi and relevant query rel-x-qj, whereas below the line it indicates the Jaccard match between the same description and non-relevant query non-rel_x_qj.

The goal of adaptation was to elevate G_g but reduce G'_g.

Figure 4--Matching of descriptions with relevant and non-relevant queries

Document	Change in overall average matching score from generation 1 to generation 40: relevant queries	Non-relevant queries
Doc 1	10.05	4.14
Doc 12	7.61	1.81
Doc 17	10.83	-5.10
Doc 18	13.11	8.51
Doc 19	8.79	4.75
Doc 21	9.11	0.96
Doc 22	10.38	0.08
Doc 23	8.01	-8.18
Doc 25	10.81	-3.29
Doc 27	8.06	6.87
Doc 28	8.51	11.43
Doc 30	9.79	3.05
Doc 32	11.12	2.94
Doc 33	7.56	5.41
Doc 34	9.11	3.48
Doc 35	10.69	1.91
Doc 36	9.17	1.69
Doc 7	5.62	-5.82
Avg.	9.35	1.92
S.D.	1.62	4.89

Data expressed in units of Jaccard's score * 100.

The pair of table entries in a row (like 10.05 and 4.14 in row 1) indicate the intentional and inadvertent improvement, respectively. That is, after Document-1 was redescribed for 40 generations, the overall average matching score relative to its relevant queries was intentionally elevated by 10.05 Jaccard points; similarly, the same redescription inadvertently increased document-1's overall average matching score 4.14 points relative to its non-relevant queries.

Table 1--Increase in overall average matching for non-relevant queries versus relevant queries

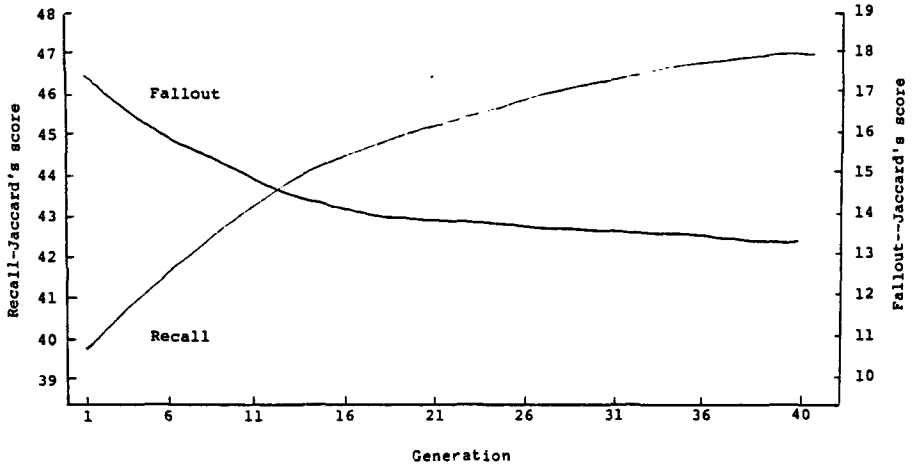

Figure 5--Recall-fallout improvement-all documents combined

Recall curve shows improving similarity between document descriptions and the fixed set of relevant queries; fallout curves shows that there is a simultaneous reduction in similarity between the same document descriptions and the non-relevant queries.

	RECALL			FALLOUT		
	Gen 1	Gen40	%Chng	Gen 1	Gen40	%Chng
Doc 1	36.03	42.86	18.96	20.07	14.47	-27.90
Doc 12	44.45	50.59	13.81	17.83	7.69	-56.87
Doc 17	42.19	52.53	24.51	17.12	11.05	-35.36
Doc 18	39.36	50.58	28.51	21.08	25.87	+22.72
Doc 19	41.12	47.33	15.10	18.83	17.58	- 6.64
Doc 21	43.01	52.45	21.95	18.00	16.04	-10.89
Doc 22	33.45	40.09	19.85	18.11	13.87	-23.41
Doc 23	31.81	39.98	25.68	12.92	4.28	-66.87
Doc 25	54.21	64.43	18.85	13.72	8.33	-39.29
Doc 27	37.92	46.65	23.02	17.65	13.25	-24.93
Doc 28	28.06	30.23	7.73	19.34	14.52	-24.92
Doc 30	48.15	57.72	19.88	16.88	18.45	+9.30
Doc 32	47.36	57.09	20.54	16.69	16.81	+0.72
Doc 33	39.95	44.29	10.86	20.29	13.75	-32.23
Doc 34	36.80	43.95	19.43	18.25	16.16	-11.45
Doc 35	39.83	47.64	19.61	17.88	13.03	-27.13
Doc 36	31.23	37.99	21.65	14.75	8.53	-42.17
Doc 7	36.66	41.68	13.69	16.35	8.31	-49.17
Average	39.53	47.12	19.09	17.54	13.44	-24.81

This table indicates the initial (pre-adaptation) level of association between a document and its relevant queries and its non-relevant queries, as well as final (post-adaptation) levels of the same measures. For doc 1, for example, we see that document redescription caused the average Jaccard's match between relevant queries and document descriptions to rise from a Jaccard's score of 36.03 (before adaptation) to a Jaccard's score of 42.86 (18.96% improvement). The same document redescription resulted in the average match between doc 1's non-relevant queries and document descriptions dropping from a Jaccard's score of 20.07 to a Jaccard's score of 14.47 (a 27.90% improvement).

Table 2--Recall-fallout improvement

Appendix A

The genetic (adaptive) algorithm operates on a set of objects, each of which is performing a similar task. The algorithm replaces this set of objects with another set, then another, and so on. The replacement, described below, attempts to produce new sets of objects in each succeeding "generation" which, on a whole, are more fit (perform the designated task better).

The genetic algorithm repeats the two-step process already outlined:

Repeat
1) Measure the performance (worth) of each competing object in a (fixed-size) set.
2) Replace the set of objects:
First, throw away the current set of objects.
Then, build new objects out of old object parts, using more parts of the objects with higher worth. Each of these new objects will likely be different than all objects in the previously discarded set.

Until some criterion is attained.

Figure 2, in the text, helps explain the details of the algorithm as used in this study. Each of the generation-g descriptions of document-x shown is really a binary document vector. For example, we might have:

		T_1	T_2	T_3	T_4	.	.	T_k	
desc_x_g1	=	< 1	1	1	0	.	.	0	>
.					.				
.					.				
desc-x-gN	=	< 0	1	1	0	.	.	0	>

where each of T_1 through T_k is a subject term (actually, phrase) that is either being employed in describing a document (1) or is not (0).

Both of the steps in the algorithm above are now more completely explained.

1) Measure performance of competing objects

The Average Matching Score for each description is indicated in the right most column of Figure 2. This measures how well each competing description "performs" (matches, on average, the M relevant queries for this document). We call this a description's "**fitness**".

2) Replacement of the set of objects

a) Relative Fitness:
Calculate, for each description, desc_x_gi,
Relative_Fitness (desc_x_gi) (1 <= i <= N).
Relative_Fitness(desc_x_gi) = Avg Matching Score (desc_x_gi)/F, where

$$F = (1/N) * \sum_{m=1}^{N} \text{Avg Matching Score (desc\_x\_gm)}$$

b) Reproduction:
Create Relative_Fitness (desc_x_gm) copies of desc_x_gm (1 <= m <= N)
Treat fractional relative fitnesses stochastically.
Discard generation g descriptions.

c) Cross_over:

Randomly partition this newly created set of N descriptions into floor (N/2) pairs (plus a single remaining description if N is odd).

For each pair, j, pick a random cross-over point, p_j, $1 <= p_j <= k - 1$ (k the length of the vector).

Form the generation g + 1 set of document descriptions as follows (set initially empty):

Add to set:
initial (desc-pair j_1) + final (desc-pair j_2)
initial (desc-pair j_1) + final (desc-pair j_1)
where
desc-pair j_1 and desc-pair j_2 are the pair of document descriptions in the j-th pair;
initial (desc-pair j_t) = first p_j positions in vector desc-pair j_t (t = 1,2)
final (desc-pair j_t) = last $(k - p_j)$ positions in vector desc-pair j_t (t = 1,2)
+ = string concatenation

For odd n, remove a randomly chosen description from the set just generated. Pair it with the as yet unpaired description. Apply cross over to this additional pair and place this newly created pair into set.

For instance, if a copy of desc_x_g1 and a copy of Desc_x_gN are chosen to be paired, and p_j is randomly selected to be 3, we see

Before crossover							After crossover						
T1	T2	T3	T4	.	.	T_k	T1	T2	T3	T4	.	.	T_k
< 1	1	1	0	.	.	1	< 1	1	1	0	.	.	0 >
< 0	1	1	0	.	.	0 >	< 0	1	1	0	.	.	1 >

$p_j = 3$

The new set of document-x descriptions would replace those in Figure 2, and the entire adaptive process would be repeated.

Note: Figure 4 presents a slightly more complicated situation, differing in its calculation of relative fitness. There, the **fitness** of any description depends on both its "recall" fitness (similarity to relevant queries) and its "fallout" fitness (dissimilarity to non-relevant queries).

That is, in Figure 4, the **fitness** of a document description, say desc_x_gi, would be equal to:

(1) Avg Recall Matching Score (desc_x_gi) +
wt * (G_g - [Avg Fallout Matching Score (desc_x_gi) - G_g])

Three observations pertain to this formula.

1) The first addend, Average Recall Matching Score (desc_x_gi), reflects the description's similarity to relevant queries.

2) The second addend reflects the description's dissimilarity to non-relevant queries. The term the G'_q-[Avg Fallout Matching Score (desc_x_gi) - G_g] is exactly the same magnitude above G'_q as Avg Fallout Matching Score (desc-x-gi) is below it. This "inversion" is necessary so that descriptions good at matching relevant queries and descriptions good at not matching non-relevant queries both contribute in an "above average" fashion to the overall fitness the description. (That is, descriptions which are quite dissimilar to non-relevant queries should contribute "fallout fitness" values greater than G'_q.)

The **relative fitness** of desc_x_gi was calculated to be:

$$\text{fitness (desc\_x\_gi)}/(1/N \sum_{j=1}^{N} \text{fitness (desc\_x\_gj)})$$

3) The weight, wt, in expression (1) was employed to balance the differing effects of a description's Avg Recall Matching Score (recall fitness) and "inverted" Avg Fallout Matching Score (fallout fitness) on its overall relative fitness. Some experimentation indicated a weight of 0.50 was appropriate to cause G_g to rise and G'_g to fall the same in succeeding generations.

REFERENCES

Brauen, T. L.
"Document vector modification,"
Chapter 24 in Salton

Gordon, Michael
"Adapting document retrieval subject descriptions to relevant user inquiries,"
Working Paper No. 383
Division of Research
Graduate School of Business Administration
The University of Michigan
Ann Arbor, Michigan, 1983

Holland, John
Adaptation in Natural and Artificial Systems
University of Michigan Press,
Ann Arbor, Michigan, 1975

Holland, John
"Escaping brittleness,"
Proceedings of the International Machine Learning Workshop
Monticello, ILL, May 1983

Oddy, R. N.
"Information retrieval through man-machine dialogue,"
Journal of Documentation, 33, 1977

Rocchio, J. J., Jr.
"Relevance feedback in information retrieval,"
Chapter 14 in Salton

Salton, G.,
"Relevance feedback and the optimization of retrieval effectiveness"
in The SMART Retrieval System--Experiments in Automatic Document Processing,
Prentice-Hall
Englewood Cliffs, N.J., 1971, Chapter 15

Salton, G., and McGill, Michael
Introduction to Modern Information Retrieval,
McGraw-Hill Book Co.,
New York, 1983

Swanson, Don
"Information retrieval as a trial-and-error process,"
in Key papers in Information Science,"
edited by Belver C. Griffith,
Knowledge Industry Publications, Inc.,
White Plains, New York, 1980

Zunde, Pranas, and Dexter, Margaret,
"Indexing consistency and quality,"
American Documentation, July, 1969