APPENDIX 1 : Miscellaneous Formulae

1 Recall/precision interpolation procedures (see A III.3.2.1)

(a) linear interpolation

the given recall/precision points nearest the required recall level are linked; the highest given recall may be linked to the point 100%R, 0%P, and the lowest given recall to the point 0%R, 100%P. (For our ordinary coordination level output the last two are not done).
The formula for precision for a specified recall value V is

$$\text{no docs below V} + ((\text{V} - \text{recall below V}) * \frac{(\text{no docs above V} - \text{no docs below V})}{(\text{recall above V} - \text{recall below V})})$$

(b) pessimistic interpolation

precision at a required recall level is the maximum precision achieved at any given higher recall point. (This also assumes that if there are several precision values for the same recall, the best of these precision values is selected). If precision above the highest given recall point is required this is taken as the precision of the last given recall/precision point. (This last is not needed for completely ranked output).

2 Salton's term discrimination function Q (see B II.1.4)

Let V_j be the set of terms (term vector) for document j, and v_{ij} be the weight (e.g. within-document frequency) of term i in document j. The centroid of all the document points in the collection N is defined as the centre of gravity or 'mean' document C, where

$$C = \frac{1}{N} \sum_{j=1}^{N} v_{ij}$$

If the similarity between pairs of documents k and j is measured by a vector matching function $r(V_k, V_j)$ (say cosine correlation, see item 3 below), where r ranges from 1 for complete similarity to 0 for complete dissimilarity, the compactness Q of the document space is

$$Q = \sum_{j=1}^{N} r(C, V_j), \quad 0 <= Q <= N$$

i.e. as the sum of similarities between each document and the centroid. The contribution of term m to the document space is then represented by $Q_m - Q$, where Q_m is the compactness of the document space with m deleted. If m is a good discriminator $Q_m > Q$, i.e. $Q_m - Q > 0$; if bad $Q_m - Q < 0$.

3 Similarity and dissimilarity functions for classification and matching (see B III.2 and B IV.1.1.1)

(a) cosine correlation

let v_i and w_i be two vectors; then the similarity r_{vw} between the vectors is

$$r_{vw} = \frac{\sum_{i=1}^{t} v_i w_i}{\sqrt{\sum_{i=1}^{t} (v_i)^2 \sum_{i=1}^{t} (w_i)^2}}$$

(b) Jaccard (Tanimoto)
using the notation of (a)

$$r_{vw} = \frac{\sum_{i=1}^{t} v_i w_i}{\sum_{i=1}^{t} v_i + \sum_{i=1}^{t} w_i - \sum_{i=1}^{t} v_i w_i}$$

more simply, for binary vectors X and Y, we define the similarity S_{XY} between the vectors as

$$S_{XY} = \frac{X \cap Y}{X \cup Y}$$

(c) normalised symmetric difference
for binary vectors X and Y, given Dice's similarity coefficient

$$S_{XY} = \frac{2|X \cap Y|}{|X| + |Y|}$$

we define the dissimilarity D_{XY} as

$$D_{XY} = 1 - \frac{2|X \cap Y|}{|X| + |Y|} = \frac{|X \Delta Y|}{|X| + |Y|}$$

When used for retrieval matching (a) is referred to as cos, (c) as dis.

4 Class definitions (see B III.2)

(a) string
starting from a given term we take the term most similar to it, then that term most similar to the latter, terminating either by looping or at a specified maximum length

(b) star
starting from a given term we take that term (or terms) most similar to it, in descending order of similarity, terminating at a specified maximum size.

(c) clique
starting from a given term we take other terms with a similarity greater than some threshold to all the current members.

(d) clump
from a given starting division of the set of terms into a putative clump A and its complement B (e.g. between a single term and the rest), we seek to minimise the 'cohesion function'

$$\frac{SAB}{SAA} * \left(\left(\frac{NA^2 - NA}{SAA} - \left(\frac{100}{P} * \frac{SAA}{NA^2 - NA} \right) \right) \right)$$

where SAB and SAA are the totals of similarity connections between the members of A and B, and between the members of A, respectively, NA is the number of terms in A, and P is a constant.

Note that as used, classes were derived by the methods for each member of the term vocabulary to be grouped. Note also that once classes are formed the internal pattern of similarity connections is disregarded, and that duplicate classes may be conflated.

5 Weighting formulae (see B III.1 and B III.2)

Given within-document frequency

f_{ij} = the frequency of term i in document j

posting frequency

p_i = the total frequency of term i over the collection

collection frequency

n_i = the number of documents containing term i

document length

d_j = the total of within-document frequencies of terms in document j

term length

t_j = the number of terms in document j

and P = the total postings in the collection

N = the total number of documents in the collection,

we define

the within-document frequency weight of a term as

$$w = f_{ij};$$

the description length weight of a term as

$$w = 10 - \text{intpt}\left(\frac{10t_j}{tot}\right) \quad \text{and} \quad w = f_{ij}\left(10 - \text{intpt}\left(\frac{10d_j}{tot}\right)\right)$$

(where tot = the next multiple of 10 above max f_j and max d_j respectively)

for term length and document length weights respectively;

the file length weight of a term as

$$w = -\log\left(\frac{n_i}{N}\right) \quad \text{and} \quad w = -\log\left(\frac{p_i}{P}\right)$$

for collection frequency and posting frequency weights respectively.
Note that in the actual implementation the former is interpreted as

$$w = -\log_2\left(\frac{n_i}{\max n_i}\right)$$

Further the program uses the function $F(n_i)$ to determine the weight of a term with collection frequency n_i where

$F(n_i) = m$, where $2^{m-1} < n_i <= 2^m$.

Now given relevance frequency

r_i = the frequency of term i in a query over the set of relevant documents for the query

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

we define

the relevance weight of a query term as

$$w = r_i \qquad (F1\tfrac{1}{2})$$

or as

$$w = \log \frac{\left(\frac{r_i}{R}\right)}{\left(\frac{n_i}{N}\right)} \qquad (F1)$$

$$w = \log \frac{\left(\frac{r_i}{R}\right)}{\left(\frac{n_i - r_i}{N - R}\right)} \qquad (F2)$$

or as

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

or as

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

Note that these are the retrospective versions of the formulae; when the formulae are to be applied predictively we add 1 to F½, and to the various components of the other formulae as follows:

$r_i + ½$, $n_i - r_i + ½$, $R - r_i + ½$, $N - n_i - R + r_i + ½$,
$R + 1$, $N - R + 1$, $n_i + 1$, $N - n_i + 1$,
$N + 2$.

The treatment for special cases in the retrospective application of F1 - F4 is given in the following table:

Case	Definition	Documents in which term occurs	Functions to which case applies	Implications of case for document when term is: Present	Absent
A	r = 0	Non-relevant only	F1,F2,F3,F4	Bad	Indifferent
B	n-r = 0	Relevant only	F2,F4	Good	Indifferent
C	R-r = 0	All relevant and some others	F3,F4	Indifferent	Bad
D	N-n-R+r = 0	Some relevant and all others	F4	Indifferent	Good
E	n-r = 0 R-r = 0	All relevant and no others	F4	Good	Bad
F	r = 0 N-n-R+r = 0	No relevant and all others	F4	Bad	Good

"Bad" means that the document should never be retrieved, i.e. should be at bottom rank;
"Good" means that the document should always be retrieved, i.e. should be at top rank;
"Indifferent" means that the document should be unaffected, i.e. should be at the rank determined by its other terms.
Case E combines B and C; case F, A and D.
Cases C through F apply to functions F3 and F4, in which term absence is explicitly recognised.