VIII. An Experiment in Automatic Thesaurus Construction

R. T. Dattola and D. M. Murray

Abstract

A method is presented for the automatic construction of thesauruses used in information retrieval systems. The construction algorithm is based on the concept-concept associations displayed in a sample document collection.

1. Introduction

Information retrieval systems often use a thesaurus look-up to determine the information content of

a) documents put into the system and
b) requests for information from the system. [1]

With respect to documents, the look-up reduces the written text to a set of concept numbers representing the keywords, phrases, and ideas of the text. With respect to requests for information, the thesaurus look-up expands a query by assigning to it concept numbers which represent more general ideas than those in the original query.

Thesaurus construction is often performed by hand or by semi-automatic methods [2]. Hand preparation is time-consuming and relies on human judgment to determine the desired thesaurus classes. Semi-automatic methods require less intellectual attention, but are also in need of human attention. A fully automatic method is desirable, since it would

a) provide rapid construction,
b) form thesaurus classes strictly on the basis of information in the document collection under consideration, and
c) apply easily to a wide range of subject areas. [4]

Naturally, the evaluation of a thesaurus is based on its performance when used in information searches. In its construction, the following criteria should ideally be followed:

a) closely related pieces of information should be assigned the same concept number;

b) the number of thesaurus classes should be significantly smaller than the number of original concepts;

c) the number of concepts appearing in more than one thesaurus class should be small; and

d) the concepts in a thesaurus class should be homogeneous; i.e. they should all occur in approximately the same number of documents.

In the present study, a document collection in a single subject area is taken as a sample vocabulary. The vocabulary is represented by previously assigned concept numbers with their associated weights. Concept-concept association techniques are then used to derive the thesaurus classes. The principle behind these techniques is co-occurrence — concepts which occur together often enough may be replaced by a single concept (a concept class).

2. The Construction Algorithm

A thesaurus is constructed in four steps:

a) formation of subcollections of documents by clustering;

b) formation of initial classes;

c) formation of merged classes by combining related initial classes;

d) formation of final thesaurus classes by eliminating merged classes that are subsets of each other.

A) Clustering the Document Collection

Rocchio's Clustering Algorithm [5] is used to divide the original document collection into subcollections of most similar documents. These subcollections contain many closely related concepts, and hence represent very broad concept classes.

Table 1 summarizes the clustering results for the 82-document ADI collection and the 200-document Cranfield collection.

B) Formation of Initial Classes

In this step, a set of initial concept classes is formed for each subcollection.

Let C denote the binary concept-document matrix constructed from each subcollection, where C consists of row vectors C_i that specify the documents in which concept i occurs. Then for any concepts i and j, a similarity coefficient S_{ij} is computed by correlating C_i with C_j. A concept-concept similarity matrix S is produced by computing these coefficients between each pair of concepts.

Several functions may be used to compute the elements of S, the most desirable ones producing a symmetric matrix with the magnitude of each element between 0 and 1. [1]

Let L_i be the number of documents in which concept i occurs; L_j, the number of documents in which concept j occurs; and N, the number of documents in which both occur. Then the following correlation functions may be used to compute S:

1) cosine: $S_{ij} = \frac{N}{\sqrt{L_i \cdot L_j}}$

2) Tanimoto: $S_{ij} = \frac{N}{L_i + L_j - N}$

3) overlap: $S_{ij} = \frac{N}{\text{Min}(L_i, L_j)}$

By applying a cut-off value k to the elements of S, the similarity matrix may be interpreted as a binary connection matrix for a graph G. The concepts form the node vector, and node i (concept i) is connected to node j (concept j) if and only if $S_{ij} \geq k$.

The initial concept classes of the subcollection are specified by the <u>maximal complete subgraphs</u> of G. [6] Given the node vector, the degree vector of these nodes, and the corresponding connection matrix, the algorithm for finding the maximal complete subgraphs of G is as follows:

1) pick the first node i which does not occur in any previous initial class and use it as the start of a new initial class;

2) for each node j connected to node i,
   a) test if the degree of node j is greater than the number of nodes in the new initial class; and

   b) test if node j is connected to all other nodes in the new class;

ADI Collection			Cranfield Collection		
(82 Documents, 601 Concepts)			(200 Documents, 2628 Concepts)		
Sub-Collection Number	Number of Documents	Number of Concepts	Sub-Collection Number	Number of Documents	Number of Concepts
1	16	167	1	17	635
2	10	118	2	12	300
3	9	75	3	10	480
4	17	135	4	12	406
5	13	117	5	7	291
6	13	126	6	15	460
7	9	86	7	15	525
			8	7	273
			9	7	271
			10	6	289
			11	7	281
			12	7	204
			13	9	325
			14	12	394
			15	15	516
			16	12	466
			17	8	310
			18	6	221
			19	14	516
			20	9	367
			21	6	218
			22	9	445
			23	9	425

Number of documents in more than one subcollection: 5

Number of documents in more than one subcollection: 31

Cluster Hierarchy

Table 1

c) if both a) and b) hold, then add node j to the new initial class;

3) repeat steps 1) and 2) until every node occurs in at least one initial class.

As an example of the formation of initial classes, consider the concept-document matrix in Fig. 1(a). Using the cosine correlation, a concept-concept similarity matrix S is constructed as shown in Fig. 1(b). Fig. 2(a) illustrates the different graphs produced by varying the cut-off value k. Finally, the resulting initial classes are shown in Fig. 2(b) for each value of k.

C) Formation of Merged Classes

If every concept in the given collection occurs in only one subcollection, then the initial classes represent the final thesaurus classes. However, it is very probable that many concepts occur in several subcollections, possibly resulting in duplicate or very similar initial classes. These similar initial classes are combined into merged classes.

Let C' denote the class-concept matrix formed from all the subcollections. Then C' consists of row vectors C_i' that specify the concepts contained in initial class i. Following the same procedure that was used to produce the initial classes, a class-class similarity matrix S' is formed. Each element S_{ij}' of S' is a measure of the similarity between class i and class j. As before, a cut-off value k is applied to the elements of S', allowing S' to be interpreted as a binary connection matrix for a graph G'. The maximal complete subgraphs of G' are the merged classes.

	D_1	D_2	D_3	D_4	D_5	D_6
C_1	1	0	0	0	0	0
C_2	1	0	1	0	0	0
C_3	1	0	1	0	0	0
C_4	0	1	0	1	1	1
C_5	0	1	0	1	0	1
C_6	0	1	0	1	0	0
C_7	0	0	1	0	0	0
C_8	0	0	0	1	0	0

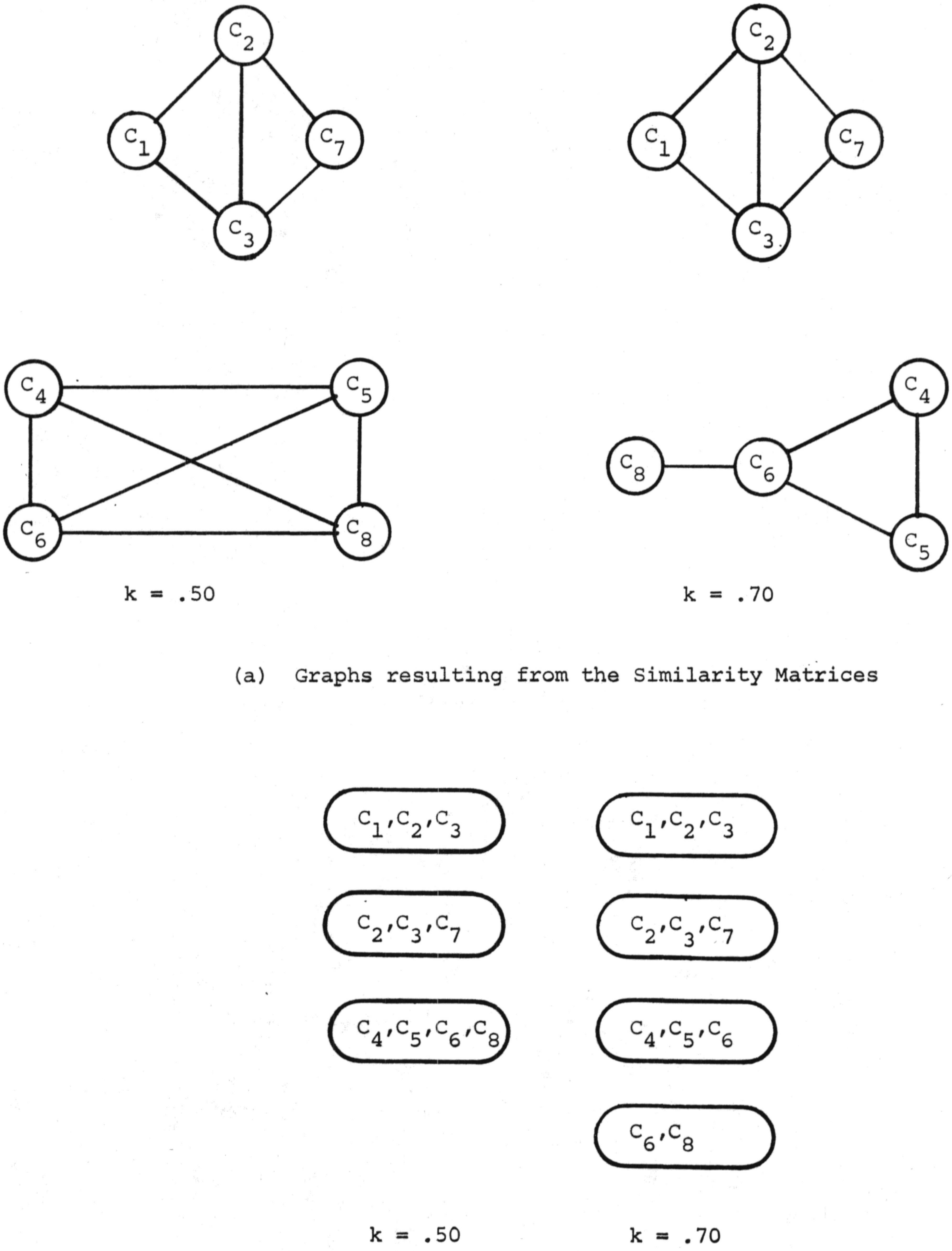
(a) Concept-Document Matrix

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	1.0	.70	.70	0	0	0	0	0
C_2	.70	1.0	1.0	0	0	0	.70	0
C_3	.70	1.0	1.0	0	0	0	.70	0
C_4	0	0	0	1.0	.86	.70	0	.50
C_5	0	0	0	.86	1.0	.81	0	.57
C_6	0	0	0	.70	.81	1.0	0	.70
C_7	0	.70	.70	0	0	0	1.0	0
C_8	0	0	0	.50	.57	.70	0	1.0

(b) Concept-Concept Similarity Matrix Cosine Correlation

Construction of Concept - Concept Similarity Matrix

Fig. 1



(a) Graphs resulting from the Similarity Matrices

(b) Complete Subgraphs

Derivation of Initial Concept Classes for one Subcollection

Fig. 2

D) Formation of Final Classes

The final thesaurus classes consist of merged classes, omitting those that are subsets or duplicates of other merged classes.

The algorithm used to construct the merged classes sometimes results in the generation of merged classes that are subsets of others. For example, Fig. 3 illustrates the formation of merged classes for a typical set of initial classes. Using the overlap correlation, $S'_{12} = 1$, $S'_{13} = 1$, and $S'_{23} = 0$. Therefore, for $k=1$, two merged classes are formed; the first consists of initial classes 21 and 52, and the second consists of initial classes 21 and 83. However, both merged classes contain the same original concepts; hence, one of them should be eliminated.

Let C" denote the class-concept matrix formed from all the merged classes. Then C" consists of row vectors C''_1 that specify the concepts contained in merged class i. Following the same procedure used to form the initial and merged classes, a similarity matrix S" is computed, and then a graph G" is formed. The maximal complete subgraphs of G" represent the final classes.

If broader concept classes are desired, the final classes can be treated as merged classes and the above step can be repeated as often as desired. Naturally, the lower the cut-off value k, the broader the final classes. However, in very broad concept classes, some of the concepts might have little or no relation to one another. On the other hand, if the number of final classes is nearly as large as the number of original concepts, then an attempt should be made to combine some of the final classes. [7] Although the final evaluation of a thesaurus is determined by its performance when used in information searches, the four principles

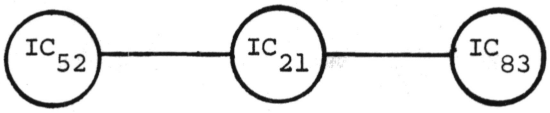
$$\begin{array}{c|cccc} & C_5 & C_{17} & C_{29} & C_{31} \\ \hline IC_{21} & 1 & 1 & 1 & 1 \\ IC_{52} & 1 & 1 & 0 & 0 \\ IC_{83} & 0 & 0 & 1 & 1 \end{array}$$

C = concept   IC = initial class

(a) Class-Concept Matrix C'

$$\begin{array}{c|ccc} & IC_{21} & IC_{52} & IC_{83} \\ \hline IC_{21} & 1 & 1 & 1 \\ IC_{52} & 1 & 1 & 0 \\ IC_{83} & 1 & 0 & 1 \end{array}$$

(b) Class-Class Matrix S'



(c) Graphs resulting from S', k = 1.0

IC_{21}, IC_{52} IC_{21}, IC_{83}

(d) Merged Classes

Formation of Merged Classes

Fig. 3

mentioned in section 1 should also be kept in mind to aid in the thesaurus construction.

3. Evaluation

The evaluation of a thesaurus for information retrieval operations is based primarily on the results of information searches using thesaurus lookup. However, the results are dependent on the characteristics of the thesaurus classes; hence, the classes are subject to an independent evaluation.

A) Evaluation of the Classes

Initial classes are composed of the similar concepts in each subcollection. Here, similarity is related to the fact that the concepts occur and do not occur in the same documents. The overlap correlation function measures similarity only on the basis of overlap between concepts, and therefore, is not used in the formation of initial classes. For example, given $c_i = (1,1,0,0,)$, $c_j = (1,1,0,0,)$, and $c_k = (1,1,1,1)$, then $S_{ij} = 1.0$ and $S_{ik} = 1.0$ for the overlap function. On the other hand, $S_{ij} = 1.0$ for the cosine and Tanimoto functions, but $S_{ik} = .71$ and $.50$ respectively. Thus, these two functions measure similarity on the basis of co-occurrence and are, therefore, of significant interest.

In order to evaluate the formation of initial classes within a subcollection, let k be the chosen cut-off value; N, the number of concepts in the subcollection; L, the number of classes formed, and M, the number of concepts appearing in more than one class. Then, define the overlap ration, class ratio, and class coefficient as follows:

1) overlap ration = M/N
2) class ratio = L/N, and
3) class coefficient = 100 (M/N) (L/N)

The class coefficient is used as a single evaluation measure for the initial classes formed from the subcollection. Because it is desirable that both the overlap ratio and the class ratio be small, it follows that the class coefficient should be small. However, if each concept were put into its own class, the overlap ratio and class coefficient would be 0, and the class ratio 1. Therefore, the three evaluation measures are best considered in conjunction with each other.

Table 2 and Fig. 4 give the values of the three evaluation measures for various cut-offs and correlation functions. Three subcollections from the ADI collection are used for comparison.

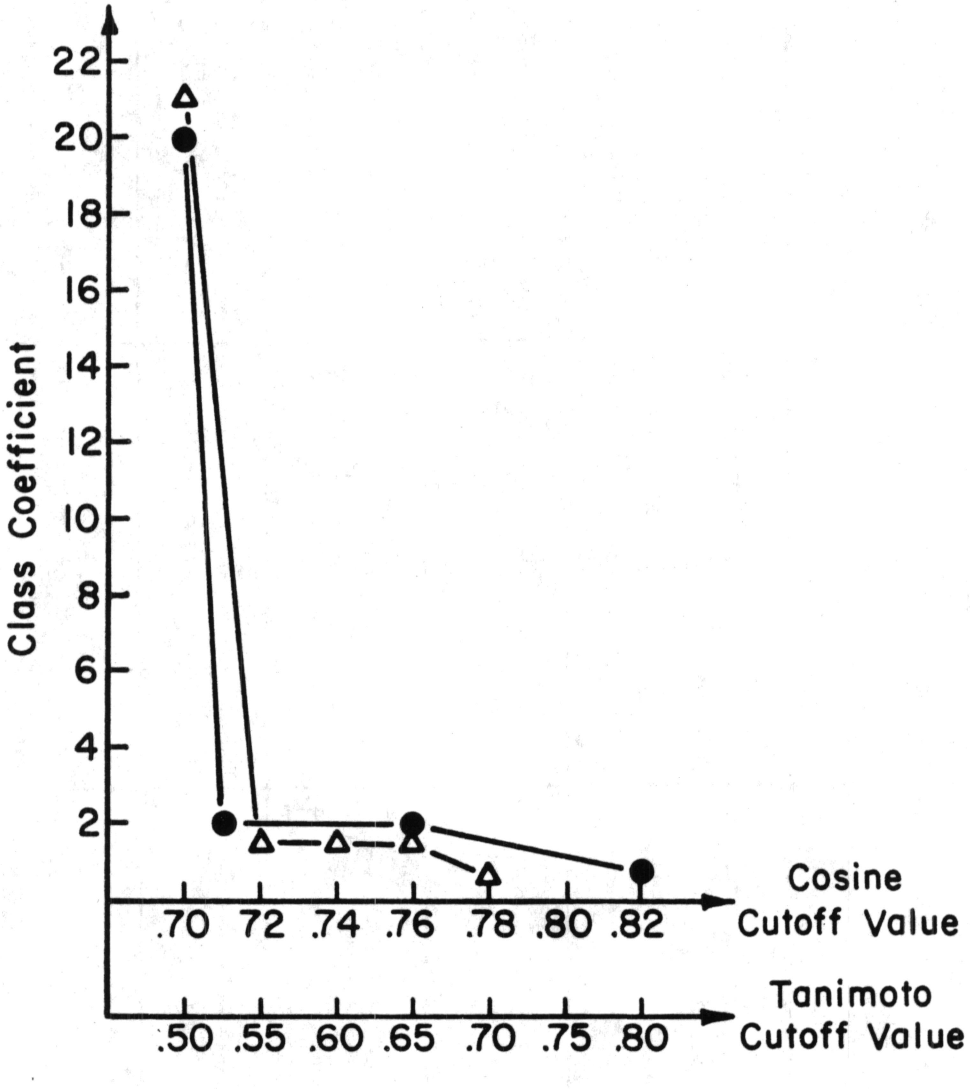
Both the cosine and Tanimoto's function give very similar results. However, the cosine function is used in the initial document clustering and in the retrieval searches. Therefore, to provide consistency, it is also used in the formation of the initial classes.

The large difference in the class coefficient between the cut-off values of .70 and .71 (cosine) is explained by the general nature of the subcollections. There are many concepts which occur in only one document. Correlating one of these concepts with any other concept in the subcollection yields one of the following values: 1, $1/\sqrt{2}$, $1/\sqrt{3}$, ... , 0. The .71 cut-off value permits graph connections only between this concept and other concepts in the <u>same</u> document which occur only <u>once</u> in the subcollection. The .70 cut-off value permits these connections and connections with concepts which appear twice in the subcollection.

Sub-Collection	Correlation Function	Cutoff	Overlap Ratio	Class Ratio	Class Coefficient
2	Cosine	.70	.915	.263	24.0
		.71	.034	.280	.95
		.76	.034	.300	1.01
		.82	0	.311	0
	Tanimoto	.50	1.03	.263	27.2
		.55	.034	.280	.948
		.60	.025	.280	.711
		.65	.034	.297	1.01
		.70	0	.331	0
4	Cosine	.70	.484	.325	15.8
		.71	.095	.365	3.48
		.76	.095	.381	3.62
		.82	.016	.444	.705
	Tanimoto	.50	.556	.317	17.6
		.55	.095	.365	3.48
		.60	.095	.365	3.48
		.65	.078	.381	3.02
		.70	.016	.444	.705
6	Cosine	.70	.593	.341	20.2
		.71	.007	.400	.296
		.76	.007	.400	.296
		.82	0	.407	0
	Tanimoto	.50	.593	.326	19.3
		.55	.007	.400	.296
		.60	.007	.400	.296
		.65	.007	.400	.296
		.70	0	.407	0

Evaluation Statistics
3 ADI Subcollections

Table 2



Evaluation Statistics

(Averaged over three of the ADI Subcollections)

Fig. 4

Because of the smaller class coefficient, the cut-off value .71 is used. Any higher value does not significantly improve any of the evaluation measures and destroys some of the useful relations in the subcollection. An added factor to be considered is that for concepts appearing only once, the only graph connections possible are to other concepts appearing once (in the same document). Such concepts may be removed from the concept-document matrix and placed in an initial class before the computation of the similarity matrix. Hence, a saving results in computation.

Instead of combining concepts which occur only once into a single concept class, each of these concepts can be treated as an individual concept class. In order to avoid confusion, the thesaurus constructed by this method will be known as THS 2, and the thesaurus constructed by combining concepts which occur only once will be called THS 1.

The merged and final classes are combinations of closely related initial classes. Therefore, it is always desirable to combine initial classes which are subsets of each other. The overlap correlation function, measuring similarity only on the basis of co-occurrence of concepts, gives a correlation value of 1.0 in such cases. For this reason, the overlap function is used in the formation of the merged and final classes. Fig. 5 gives some statistics on THS 1 and THS 2.

B) Retrieval Evaluation

To evaluate the retrieval performance of an automatic thesaurus, three information searches are used — one search with a document and query collection before the thesaurus lookup; one search after the lookup in a manual thesaurus, and one search after the lookup in the automatic thesaurus. By comparing the precision and recall statistics for all three searches, the

	THS 1	THS 2
Total number of concept classes	156	289
Avg. number of concepts per class	3.9	1.4
Number of concepts appearing in more than one concept class	167	42
Number of concepts appearing in more than six concept classes	3	0
Avg. number of classes per concept	2.1	1.2
Avg. standard deviation (S.D.) of concept frequencies per concept class	3.9	1.4

$$\text{avg. S.D.} = 1/n \sum_{i=1}^{n} (1/m \sum_{j=1}^{m} |A - f_j|) \quad \text{where,}$$

n = total number of concept classes

m = number of concepts in concept class j

A = avg. frequency of concepts in concept class i

f_j = frequency of concept j in concept class i

Statistics on Automatic Thesauruses

Fig. 5

effectiveness of the automatic thesaurus may be decided.

The thesaurus collections are formed by treating each document or query independently. For each concept-weight pair (n,w), the thesaurus classes — $N_1,N_2,\ldots,N_k$ — corresponding to n are determined by a table lookup procedure. The concept-weight pairs added to the new document(query) are $(N_1, w/k)$, $(N_2, w/k),\ldots,$ $(N_k, w/k)$. If k is greater than 6 for a given concept n, the concept is dropped from the thesaurus. This is done because of space limitations, but these concepts would probably have very small weights anyway since the weight is divided by k. At the end of the lookup, concept pairs with duplicate concept numbers are eliminated. The duplicates are replaced by a single concept-weight pair whose weight is the sum of the weights in the duplicates.

In the ADI collection, the lookup procedure produces a document and query collection with more concepts per document than in the original. The weights associated with these concepts are smaller than before, although the sum of the weights in both collections is nearly equal for THS 1.

4. Analysis of Results

The results of the search evaluation for the ADI thesauruses are given in Fig. 6. The weighted cosine function is used to match the queries against the documents. Given query i and document j, the correlation is defined as follows:

$$s_{ij} = \frac{\sum_{k=1}^{t} q_k \cdot d_k}{\sqrt{\sum_{k=1}^{t} (q_k)^2 \cdot \sum_{k=1}^{t} (d_k)^2}}$$

where q_k is the weight of concept k in query i, d_k is the weight of concept k in document j, and t is the total number of concepts.

Because the original ADI collection is a manual thesaurus, the automatic thesauruses constructed from this collection are actually superthesauruses. However, both THS 1 and THS 2 give better results than the original manual thesaurus. Two evaluation functions that are useful for comparing the retrieval results of a given query using different thesauruses are the <u>normalized recall</u> and the <u>normalized precision</u>. Specifically,

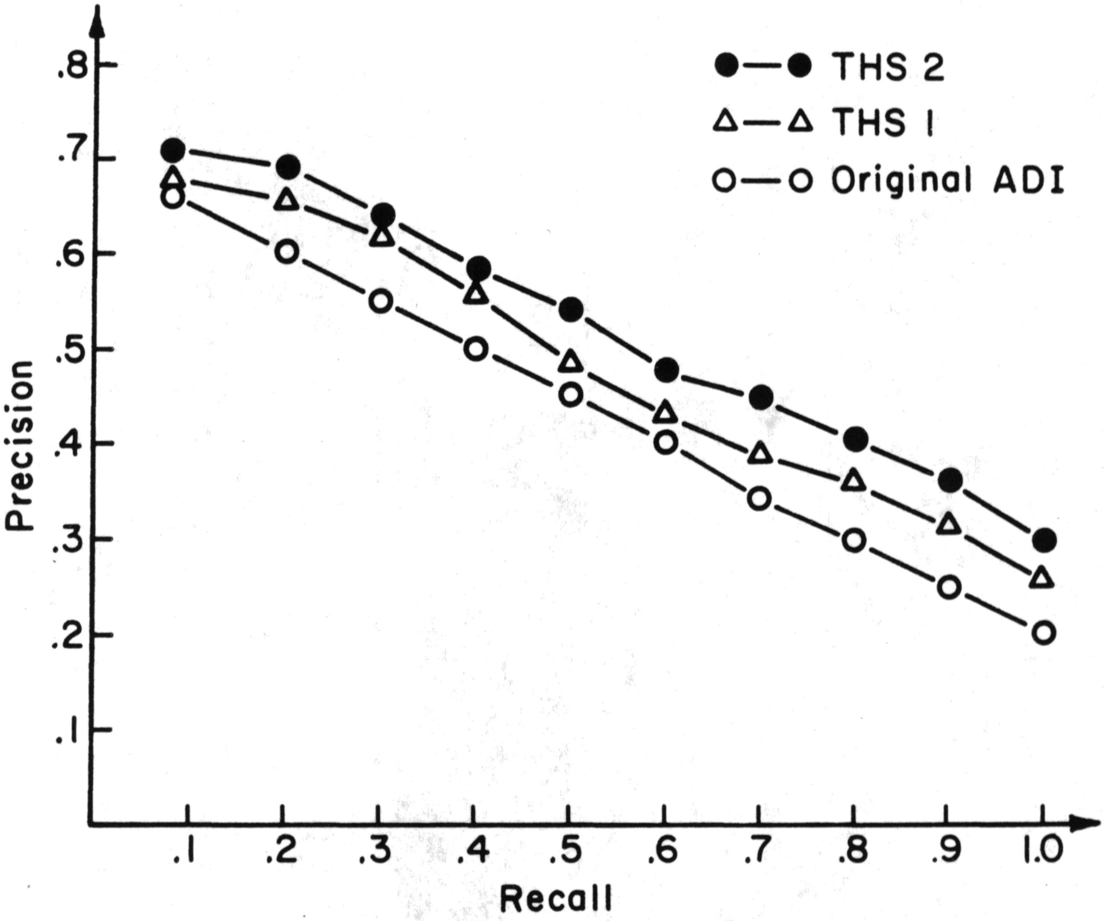
$$\text{N.P.} = 1.0 - \frac{\sum_{i=1}^{n} \ln r_i - \ln n!}{\ln \binom{N}{n} - \ln n!} \text{ , and}$$

$$\text{N.R.} = 1.0 - \frac{1}{(N-n).n} \sum_{i=1}^{n} (r_i - i)$$

where N is the total number of documents, n is the number of relevant documents, and r_i is the rank of the i^{th} relevant document. The normalized recall and precision values for the three ADI searches are given in Table 3.

Although THS 2 gives the best results overall, there are several queries where the original thesaurus is best and several queries where THS 1 is best. A closer inspection of the results indicates the following conclusions:

a) the amount of overlap between concept classes of a manual thesaurus such as the ADI can be increased by automatic procedures to produce better results;



Evaluation Results

Fig. 6

	Original		THS 1		THS 2	
Query	N.R.	N.P.	N.R.	N.P.	N.R.	N.P.
1	.94	.70	.96	.88	.92	.62
2	.95	.69	.72	.33	.82	.49
3	.98	.88	.96	.87	.98	.90
4	.82	.67	.91	.74	.94	.78
5	.98	.93	.92	.62	.97	.80
6	.84	.67	.96	.83	.87	.70
7	.90	.73	.74	.36	.75	.41
8	.91	.53	.96	.69	.99	.84
9	.50	.17	.91	.56	.72	.36
10	.64	.43	.67	.58	.59	.42
11	.56	.46	.55	.45	.55	.42
12	.80	.56	.93	.75	.86	.66
13	.62	.37	.65	.31	.66	.38
14	.78	.46	.92	.57	.88	.71
15	.70	.51	.65	.41	.81	.60
16	.78	.43	.83	.68	.78	.51
17	.77	.55	.97	.78	.93	.69
18	.88	.79	.92	.83	.92	.83
19	.78	.58	.81	.55	.81	.57
20	.76	.44	.90	.59	.90	.81
21	.96	.87	.99	.97	.99	.97
22	.78	.57	.85	.64	.87	.72
23	.99	.84	1.00	1.00	1.00	1.00
24	1.00	.99	1.00	1.00	1.00	1.00
25	.95	.86	.98	.91	1.00	1.00
26	.81	.37	.69	.26	.54	.17
27	.59	.63	.62	.65	.64	.64
28	.87	.74	.85	.72	.94	.83
29	.75	.31	.41	.12	.52	.16
30	.57	.48	.67	.57	.71	.57
31	.52	.32	.62	.40	.58	.35
32	1.00	1.00	.95	.80	.98	.89
33	.65	.34	.96	.83	.89	.57

Table 3

	Original		THS 1		THS 2	
Query	N.R.	N.P.	N.R.	N.P.	N.R.	N.P.
34	.72	.54	.71	.56	.73	.49
35	.94	.89	.85	.63	.95	.89
Average over all queries	.80	.61	.83	.64	.83	.65

Results of Retrieval

(continued)

Table 3

b) concepts which occur in only one document within a group should be treated as individual concept classes as in THS 2;

c) the concepts within a thesaurus class should be homogeneous; i.e., they should all occur in approximately the same number of documents;

d) when expanding a query or document by a thesaurus, the concept class weights should be divided by the number of concept classes in which a concept appears.

A) Overlap

Because the original ADI collection is already a thesaurus, THS 1 and THS 2 have in effect combined many of the original concept classes, thereby producing more overlap between the classes. For example, in query 15, document 67 (a relevant document) is ranked 68th using the original thesaurus and 26th using THS 2. As shown in Fig. 7, in the original thesaurus, only one out of eight concepts in the query also occurred in the document, while in THS 2, there were five out of eighteen matches. The improvement is due to concepts 10, 22, and 104 which appear in query 15 but not in document 67. Specifically, concept class 36 contains concept 1 and concept 10; concept class 136 contains concept 1 and 104, and concept class 203 contains concepts 9 and 22. Therefore, both document 67 and query 15 contain concept classes 36, 136, and 203 after the lookup in THS 2.

B) Unique Concepts

THS 1 combines all concepts which occur in only one document of a group into a single concept class. The disadvantage of this method is illustrated by examining, for example, query 2 and document 12 (a relevant document), which was ranked 35th by THS 1 and 26th by THS 2. This document contains the following concepts which do not appear in query 2: 18, 25, 97, 116, 154, 261, 304, 338, 399. Concept class 82 of THS 1 contains the following concepts: 25, 97, 116, 154, 338, 399. These concepts occur only in document 12 in group 4; therefore, they become a single concept class, class 82. Thus, their weights are all added together during the expansion of document 12, producing a very high weight for concept class 82. However, if one of the concepts in this class had appeared in query 2, the correlation with document 12 would have been much higher. Therefore, the presence or absence of one concept in the query makes a large difference in retrieval. When the query happens to contain one of these "unique" concepts, THS 1 usually performs better than the original thesaurus or THS 2.

C) Homogeneous Concept Classes

Another disadvantage of THS 1 is that the concept classes are not very homogeneous. Fig. 5 shows that the average standard deviation of frequency among concepts in a concept class is 3.9 for THS 1 and 1.4 for THS 2. Thus, a query containing a concept which occurs in few documents, but which is in a concept class with a concept occurring in many documents may retrieve several irrelevant documents. For example, query 5 contains concepts 1, 5, 13, 38, 94, 115, 533 with frequencies 44, 29, 10, 4, 3, 5, and 11 respectively. Concept 94 (frequency = 3) occurs in concept class 70 along with concept 67 (frequency = 13), concept 89 (frequency = 9), and concept 21 (frequency = 13). Document 46 is the only document containing concepts 21, 67, and 89. Since concept 94 maps into concept class 70, query 5 might be expected to retrieve document 46, an irrelevant document. This is exactly what happens, for document 46 is ranked 23rd using the original thesaurus and is ranked 4th using THS 1.

D) Dividing Weights

A common objection against automatic thesauruses is that they contain too much overlap between concept classes. Thus, the concepts which occur in several concept classes (which are in fact the most common concepts) do not contribute much to the thesaurus, as their weights are divided by the number of concept classes in which they occur. The evaluation results using THS 1 and THS 2 indicate that manual thesauruses do not contain enough overlap, rather than automatic thesauruses contain too much overlap. However, an argument might be raised against dividing the weights. To settle this argument, THS 2 was also evaluated without dividing the weights during the lookup. The results were much worse:

N.R. = .74, down from .83, and

N.P. = .52, down from .65.

E) Cranfield Collection

Although the results from the ADI text are encouraging, the goal is to produce an automatic thesaurus starting from a word stem thesaurus rather than a regular manual thesaurus. Since the original concepts in a stem thesaurus are not themselves manually constructed concept classes, it can be expected that many more connections exist between the original concepts than in a regular thesaurus. Thus, THS 1 constructed from the Cranfield stem thesaurus contains too much overlap between concept classes. In fact, over

300 concepts were dropped from the thesaurus because they occurred in more than six concept classes! As shown in Fig. 5, THS 2 produces much less overlap than THS 1. Unfortunately, THS 2 has not yet been constructed for the Cranfield collection; hopefully, it will give much better results than THS 1. Also, the overlap can be reduced by raising the cut-off value for the initial classes.

F) Comparison with Other Methods

The algorithm presented for the automatic construction of thesauruses defines initial concept classes only on the basis of internal similarities among concepts in the class — every concept is related to every other concept in the concept class. Other methods have been described which do not require as much internal similarity, but which also attempt to minimize the relations between concept classes. [8] In the present scheme, this is accomplished by the formation of merged and final classes, which requires the calculation of two more similarity matrices.

The division of the original document collection into subcollections permits new concepts to be added to the thesaurus as a natural extension of the construction method. The concepts to be added are placed in a new subcollection, and the initial classes generated as before. All that remains is to merge these new classes with the existing ones by using the procedure described in section 2D.

References

[1] G. Salton, M. Lesk, Information Analysis and Dictionary Construction, Report No. ISR-11 to the National Science Foundation, Section IV, Cornell University, June 1966.

[2] C. Harris, Dictionary and Hierarchy Formation, Report No. ISR-7 to the National Science Foundation, Section III, Harvard Computation Laboratory, June 1964.

[3] C. Harris, Dictionary Construction and Updating, Report No. ISR-8 to the National Science Foundation, Section VII, Harvard Computation Laboratory, June 1964.

[4] G. Blomgren, A. Goodman, L. Kelly, An Experimental Investigation of Automatic Hierarchy Generation, Report No. ISR-11 to the National Science Foundation, Section VIII, Department of Computer Science, Cornell University, June 1966.

[5] J. J. Rocchio, Harvard University Doctoral Thesis, Report No. ISR-10 to the National Science Foundation, Harvard Computation Laboratory, April 1966.

[6] C. T. Abraham, Techniques for Thesaurus Organization and Evaluation, Information Science, M. Kochen, editor, Scarecrow Press, 1965.

[7] C. C. Gotlieb, S. Kumar, Semantic Classification of Index-Terms, University of Toronto, unpublished.

[8] K. S. Jones, D. Jackson, Current Approaches to Classification and Clump - Finding at the Cambridge Language Research Unit, The Computer Journal, Vol. 10, May 1967.