EXPERIMENTS IN AUTOMATIC STATISTICAL THESAURUS CONSTRUCTION

Carolyn J. Crouch and Bokyung Yang*
Department of Computer Science
University of Minnesota, Duluth
Duluth, MN 55812

ABSTRACT

A well constructed thesaurus has long been recognized as a valuable tool in the effective operation of an information retrieval system. This paper reports the results of experiments designed to determine the validity of an approach to the automatic construction of global thesauri (described originally by Crouch in [1] and [2]) based on a clustering of the document collection. The authors validate the approach by showing that the use of thesauri generated by this method results in substantial improvements in retrieval effectiveness in four test collections. The term discrimination value theory, used in the thesaurus generation algorithm to determine a term's membership in a particular thesaurus class, is found not to be useful in distinguishing *between* thesaurus classes (i.e., in differentiating a "good" from an "indifferent" or "poor" thesaurus class). In conclusion, the authors suggest an alternate approach to automatic thesaurus construction which greatly simplifies the work of producing viable thesaurus classes. Experimental results show that the alternate approach described herein in some cases produces thesauri which are comparable in retrieval effectiveness to those produced by the first method at much lower cost.

*Current address: West Publishing Company, Eagan, Minnesota, 55123.

15th Ann Int'l SIGIR '92/Denmark-6/92
© 1992 ACM 0-89791-524-0/92/0006/0077...$1.50

INTRODUCTION

The value of a well constructed thesaurus to the effective operation of an information or document retrieval system is widely recognized. In this paper, the authors concentrate on the *global thesaurus*, wherein thesaurus classes (classes of similar or "synonymous" terms) are first generated and then used to index both documents and queries. Many experiments regarding the construction of such thesauri may be found in the literature. Early approaches relied on manual construction (e.g., Cleverdon in the Aslib Cranfield project [3]). Early Smart experiments in semiautomatic and automatic thesaurus construction [4, 5, 6] and Spark Jones' experiments in automatic keyword classification [7] made valuable contributions to the state of knowledge on the subject.

A more recent attempt to produce automatic dictionaries has centered on the relational thesaurus [8, 9]. This work is based on identifying sets of lexical or lexical-semantic relations that exist between word pairs (e.g., collocation, taxonomy, synonymy, etc.). Fox continues early work in this area with an artificial intelligence-based approach relying on the CODER relational lexicon [10, 11]. Other researchers use expert system techniques to generate dictionaries [12, 13]. (These approaches, while of considerable interest, remain largely to be evaluated.)

This paper reports on the results achieved by using an algorithm for automatic thesaurus generation described in detail in [1] and [2]. This algorithm is based on an appropriate clustering of the document space and the use of the term discrimination value theory of Salton, Yang, and

Yu [14] to select terms for entry into a particular thesaurus class. It produces not a universal lexicon, such as that envisioned by Fox, but rather a thesaurus applicable only to the collection from which it was generated. Although the resultant thesaurus is domain dependent, the algorithm may be applied to any document collection, regardless of subject (as long as the system representation is based on the vector model).

The term discrimination value model itself is based on the well-known vector space model [15], wherein both documents and queries are regarded as (weighted) term vectors and the correlation between vectors is computed by means of a similarity measure, such as cosine. Although the vector model offers advantages (such as ranking of retrieved documents, clustering, and term weighting), it also has its disadvantages (such as its inability to express term relationships within the model). A practical solution to this problem was suggested by Salton, Yang, and Yu, who showed that the best document space for retrieval purposes is one which maximizes the average separation between documents in the document space. In this space, it is easier to distinguish between documents and to retrieve documents which are most similar to a given query. This is the basis of the term discrimination value model, which assigns specific roles to single terms, term phrases, and term classes.

The discrimination value of a term is defined as a measure of the change in space separation which occurs when a given term is assigned to the document collection. A *good discriminator* is a term which, when assigned to a document, decreases the space density (rendering the documents less similar to each other). A *poor discriminator*, then, increases space density. By computing the density of the document space before and after the assignment of each term, the discrimination value of the term can be determined. The terms are then ranked in decreasing order of their discrimination values into three categories: *good discriminators* (terms with [strong] positive discrimination values), *indifferent discriminators* (terms with near-zero discrimination values), and *poor discriminators* (terms with negative discrimination values). Salton *et al* suggest that good discriminators be used directly as index terms. The retrieval properties of poor discriminators can be improved by combining these terms with others to form appropriate phrases. The retrieval capabilities of the majority of terms, which are indifferent discriminators with near-zero discrimination values, can be improved by incorporating them into appropriate thesaurus classes.

The calculation of discrimination value is normally expensive. For this reason, Salton, Yang and Yu suggested that the document frequency of a term be used as an approximation to discrimination value. The document frequency, f_k, of a term k is defined as the number of documents in which k appears. According to [14], if n represents the number of documents in a collection whose terms are ranked by increasing document frequency, terms may be classified by document frequency as follows: if $f_k > n/10$, k is considered a high frequency term (a term which normally has a negative discrimination value and is a poor discriminator); if $f_k < n/100$, k is considered a low frequency term (a term with a near-zero discrimination value, an indifferent discriminator); and if $n/10 \leq f_k \leq n/100$, k is a medium frequency term (with a positive discrimination value, a good discriminator). Empirical results have shown that document frequency and discrimination value are well correlated. A recent modification of the centroid approach to the calculation of discrimination value, by El-Hamdouchi and Willet [16], has greatly reduced the time required to calculate discrimination value, thus making it possible to use discrimination value instead of the [easily calculated] document frequency as an indicator of how that term relates to other terms in the collection (i.e., whether it should be used alone as an index term, used as a component of a phrase, or combined with other terms to become a member of a thesaurus class).

AN ALGORITHM FOR CONSTRUCTING GLOBAL THESAURI

A thesaurus is composed of a set of thesaurus classes. A thesaurus class is made up of a set of "closely related" terms--terms which are similar enough in the context of the specified collection to justify their being used together or combined to represent a single, new word type within this collection. The term discrimination value theory specifies that the terms which make up a thesaurus class must be indifferent discriminators (or low frequency terms). One cannot expect to classify terms in the low frequency domain, which have small co-occurrence frequencies and low computed similarities, into useful groupings due simply to the lack of information about them. The alternative is to cluster the document collection and subsequently to select from closely related documents terms which meet the criterion (i.e., terms with near-zero discrimination values or low frequency terms).

A clustering algorithm which produces small, tight clusters is the complete link algorithm [17, 18, 19], one of the class of agglomerative, hierarchical methods. This algorithm may be described succinctly as follows: At each stage, the similarity between all clusters is computed, the two most similar clusters are merged, and the process continues until only one cluster remains. Since the similarity between clusters is defined as the minimum of the similarities between all pairs of documents (where one document of the pair is in one cluster and the other document is in the other cluster), the resultant clusters are difficult to join and hence tend to the small and tight.

An algorithm to construct global thesauri has been previously suggested by Crouch [1, 2]. A brief description of this algorithm follows:

1. The document collection is clustered via the complete link clustering algorithm.
2. The resultant hierarchy is traversed and thesaurus classes are generated, based on three user-supplied parameters.
3. The documents and queries are indexed (augmented) by the thesaurus classes.

The characteristics of the thesaurus classes generated in step 2 are determined by these parameters:

(i) *THRESHOLD VALUE*

Application of the complete-link clustering algorithm produces a hierarchy in which the tightest clusters (i.e., those which cluster at the highest threshold values) lie at the bottom of the cluster tree. These nodes are the leaves of the tree. Consider Fig. 1. The squares represent documents and the numbers in the circles represent the levels at which the documents cluster. Documents A and B cluster at a threshold value of 0.089, D and E cluster at a level of 0.149, and document C clusters with the D-E subtree at a level of 0.077. The A-B subtree and the C-D-E subtree cluster at a threshold value of 0.029.

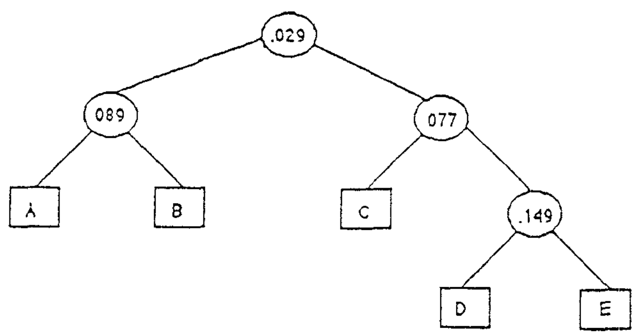

Fig. 1 A sample hierarchy generated by the complete link algorithm

Threshold value largely determines the documents from which terms are selected for inclusion in a thesaurus class. In Fig. 1, a threshold value of 0.090 would return only the D-E document cluster, since the level at which the documents cluster must be greater than or equal to the specified threshold in order for the cluster to be recognized. A threshold value of 0.085, on the other hand, would return two clusters (A-B and D-E). If a threshold value of 0.075 were specified, two clusters would be returned: A-B and either D-E or C-D-E.

(ii) *NUMBER OF DOCUMENTS PER CLUSTER*

Since both D-E and C-D-E meet the threshold value of 0.075 in Fig. 1, this parameter is needed to determine the number of documents to be included in the final cluster. In the example, a value of 2 for this parameter would insure that only the D-E cluster be returned. A value of 3 or greater would allow the entire C-D-E cluster to be returned.

(iii) *MINIMUM DOCUMENT FREQUENCY*

Once the document clusters have been selected, thesaurus classes can be formed from the low frequency terms contained in those clusters. In this case, the minimum document frequency of each term, i.e., the criterion by which terms are determined to be "low frequency," must be specified. (Using as a guideline $f_k < n/100$ as suggested in [14] may be helpful in setting this parameter.) A thesaurus class is then defined as the intersection of all the low frequency terms in that cluster.

[Alternatively, one could consider using discrimination value to determine membership in a thesaurus class. In this case, an upper bound on discrimination value might be used in lieu of minimum document frequency to determine thesaurus class membership. The intersection of those terms whose discrimination values fall beneath this bound then forms the thesaurus class. This approach is investigated in a later section (see Experiment 3).]

An important factor in retrieval effectiveness is the method by which the thesaurus classes are weighted. Previous experiments [1] have shown that the following formula is useful in computing reasonable thesaurus class weights: Let tc_wt denote the thesaurus class weight, $tc\_con\_wt_i$ denote the weight of term i in this thesaurus class, |tc_con| represent the number of terms in the thesaurus class, and avg_tc_con_wt represent the average weight of a term in the thesaurus class. Then

$$avg\_tc\_con\_wt = \frac{\sum_{i=1}^{|tc\_con|} tc\_con\_wt_i}{|tc\_con|}$$

and

$$tc\_wt = \frac{avg\_tc\_con\_wt}{|tc\_con|} * 0.5$$

Thus a thesaurus class weight is computed by dividing the average weight of a term in the class by the number of terms in the class and down weighting that value by 0.5.

THE EXPERIMENTS

Setting the Input Parameters

General comments on the input parameters may be useful before describing the individual experiments. *Threshold value* is collection-dependent. Appropriate threshold values may be determined by examining the cluster tree. This parameter largely determines the size of the thesaurus class. If the setting is too high, thesaurus classes will be generated with too few terms (i.e., the intersection of low frequency terms or indifferent discriminators will be very small). If the threshold is too low, very few classes will be generated. The *number of documents per cluster* has a lesser effect on thesaurus class size. Based on previous experiments [1], this parameter is allowed to range from 2 to 5. (Note that this parameter is always secondary to the threshold value.) *Minimum document frequency* may be set by using the guidelines suggested in [14].

Collection Weighting and Similarity Measure

When the global thesaurus has been constructed and both documents and queries augmented by the thesaurus classes, the vectors are then weighted using the facilities of the Smart system [20]. The collection weighting method used in these experiments is a variation of the inverse document frequency scheme (see [20] for a description of "atc" weighting). The similarity measure used is cosine.

Collection Characteristics

The thesaurus construction procedure described in this paper was run on four standard test collections, namely, ADI, Medlars, CACM, and CISI. The characteristics of these collections are found in Table 1. (For more complete descriptions of these collections, see [18, 21,22].)

Experiment 1

The algorithm described previously was applied to each of the four specified document collections. The results are presented in Table 2. In all cases, the membership of a term in a thesaurus class is based on minimum document frequency rather than discrimination value. The parameters yielding the best improvement in three point average (average precision at three points of recall [0.25, 0.50, 0.75], as provided by the Smart evaluation routines) are specified for each collection. The column labeled "improvement" refers to the improvement in three point average of the collection using this thesaurus (the best case) over the same collection without the thesaurus (the base case).

The number of runs made to produce each thesaurus was largely dependent on our ability to find the threshold value whose use produced the "best" thesaurus classes. Nor was the optimal value of minimum document frequency easily found; as Table 2 indicates, the guidelines of [14] were useful in only one case. Number of documents per cluster ranges from 2 to 5.

As Table 2 shows, the use of the thesaurus generates substantial improvement in each case.

Table 1

Collection Statistics

Collection	No of Documents	No of Queries	No of Terms	Mean No of Terms per Document	Mean No of Terms per Query
ADI	82	35	822	25.5	7.1
Medlars	1033	30	6927	51.6	10.1
CISI	1460	76	5019	45.2	22.6
CACM	3204	64	8503	22.5	8.7

Table 2

Improvement Using Thesaurus Generated by the Specified Parameters (Best Case)

Collection	Threshold	No of Documents	Frequency	Percentage Improvement
ADI	0.075	5	25	10.6
Medlars	0.120	3	45	17.1
CISI	0.058	4	69	7.7
CACM	0.137 (0.138)	2	30	9.6

Experiment 2

For each collection specified above, a thesaurus was generated. Each thesaurus class within the thesaurus then represents a new word type in the collection. The term discrimination value model tells us that each unique term in a collection can be classified by its discrimination value as a good, indifferent, or poor discriminator. The question that now arises is: is it possible to use discrimination value to differentiate *between* thesaurus classes--i.e., to distinguish good (or useful) thesaurus classes from poor or indifferent classes?

Our objective was to produce a more useful thesaurus by removing thesaurus classes which, although generated by our algorithm, nevertheless were not useful in improving retrieval effectiveness. (Since the indexing was done by augmentation, the removal of such classes should at least reduce the negative effects of vector expansion to some degree). Thus the discrimination values of each term in the collection (including, of course, the new thesaurus classes) was calculated via [16]. Table 3 shows the number of thesaurus classes generated (using the parameter settings in Table 2) for each collection and the number of positive and negative discriminators within each set.

We then removed from each thesaurus those thesaurus classes which may be considered poor discriminators (i.e., have negative discrimination values) according to the term discrimination value model. The resultant (reduced) thesaurus was then used to index the documents and queries. The results are shown in Table 4.

It would appear from Table 4 that the classes which were removed from the thesauri because they had negative discrimination values nevertheless added useful information in terms of retrieval. In ADI, 3 of 28 classes were removed from the thesaurus, resulting in a negligible improvement in retrieval effectiveness over the base case and a corresponding degradation over the best case (experiment 1). For Medlars, 140 negative discriminators were removed from the thesaurus, leaving 178 positive discriminators; the resultant thesaurus offers an improvement of 14 percent over the base case but is still less effective than the best case thesaurus. For CISI, 29 thesaurus classes with negative discriminators were removed, leaving 269 positive

Table 3

Number and Type of Thesaurus Classes Generated

Collection	Number	Number with Positive DVs	Number with Negative DVs
ADI	28	25	3
Medlars	318	178	140
CISI	298	269	29
CACM	438	438	0

Table 4

Improvement Using a Thesaurus Containing Only Thesaurus Classes with Positive Discrimination Values

Collection	Percentage over Base Case	Percentage over Best Case
ADI	1.3	-8.4
Medlars	14.0	-2.7
CISI	7.7	0.0
CACM	9.6	0.0

discriminators; the result is a 7.7 percent improvement over the base case and no improvement over the best case. (All of the thesaurus classes generated for the CACM collection in experiment 1 were positive discriminators.)

We also experimented by taking the set of thesaurus classes ordered by discrimination value and successively removing 5 (10, 15, ...) percent of bottom-ranked thesaurus classes until all classes with negative discrimination values had been removed. In no case were we able to improve the retrieval effectiveness of a thesaurus over our best case figures.

Experiment 3

In this experiment, we consider the result of modifying our thesaurus construction algorithm by replacing *minimum document frequency* with an upper bound on discrimination value. Using the document hierarchies and the same parameter settings (for *threshold* and *number of documents per cluster)* found in Table 2 to identify the clusters, the algorithm is modified by changing the criterion for entry of a term into a thesaurus class. In this case, *minimum document frequency* is replaced by an upper bound on *discrimination value*; i.e., thesaurus classes are formed by the intersection of those terms from the cluster whose discrimination values are greater than 0 and less than the upper bound. (The objective is to select terms having near-zero discrimination values which occur in each document of the cluster for membership in that thesaurus class.) The bound on discrimination value varies in increments of 5 percent (using 5, 10, 15, ..., percent of the qualified terms with near-zero discrimination values, extending to 100 percent [which of course includes those terms with strong positive discrimination values]). The results are found in Table 5.

Note that these figures represent the best results produced by this approach. Initially, we had hoped to construct thesaurus classes whose membership was determined only by discrimination value. It immediately became apparent that using an upper bound beneath 50 percent restricted the entry criterion to the extent that no classes were formed. The best results generated for ADI and Medlars, using all terms with positive discrimination values, were substantially less than the results of experiment 1. (Consider what occurs when this approach is applied to CACM. The best case results in 847, rather than the original 438, thesaurus classes being generated. As the upper bound on discrimination value is moved up, terms with larger frequencies become eligible to enter one or more thesaurus classes. Greater numbers of classes are produced; the classes themselves may become large. Because the correlation between discrimination value and frequency is not exact [i.e., terms with fairly low discrimination values may in fact be high frequency terms], we find no effective way to handle this problem without introducing again a limit on frequency.)

Thus our attempts to produce a thesaurus whose classes are based on the discrimination values of their component terms have not proved successful.

Table 5

Improvement Using a Thesaurus Whose Classes Are Based on Discrimination Value

Collection	Percentage over Base Case	Percentage over Best Case
ADI	8.9	-1.5
Medlars	13.9	-2.8
CISI	-16.9	-24.1
CACM	-6.7	-13.3

Experiment 4

When the thesauri produced in experiment 1 were evaluated, the results (in terms of the three point average) were averaged over all the queries in the collection. But some queries are not affected by the indexing (step 3 of the thesaurus construction algorithm). To gain a more accurate view of the effect of the thesauri on the collections, in experiment 4 we extract from each collection only those queries which are affected by indexing. This query set, the reduced set, is now used to perform a normal search (the base case). The thesaurus construction algorithm is then applied to the collection (documents and reduced query sets) and a search is performed. The results (using the parameter settings of experiment 1) are described in Table 6.

The last column in Table 6 gives an accurate and realistic view of the changes in retrieval effectiveness produced when a well constructed global thesaurus is applied to a collection.

Experiment 5

The algorithm described in this paper can be employed to produce useful global thesauri, as Table 6 indicates. But this algorithm is strongly dependent on the setting of three parameters; *threshold value* is collection-dependent and can be very difficult to set properly, and the setting of *minimum document frequency* may also prove troublesome. These parameters are used in the thesaurus construction algorithm to help identify clusters of closely related documents and then to select terms with certain characteristics to make up the thesaurus classes.

Consider another approach to identifying these document clusters. Define a low level cluster as a cluster all of whose children are documents. Our first algorithm is now revised as follows:

1. The document collection is clustered via the complete link clustering algorithm.
2. The resultant hierarchy is traversed and thesaurus classes are generated by
   -identifying each low level cluster, and
   -for each such cluster, constructing a thesaurus class by forming the intersection of all low frequency terms in the cluster.
3. The documents and queries are indexed (augmented) by the thesaurus classes.

Table 6

Improvement Using Thesaurus (Best Case) With Reduced Query Set

Collection	No of Queries Base Case	No of Queries Reduced Set	Results Using All Queries	Results Using Reduced Set
ADI	35	33	10.6	11.0
Medlars*	30	30	17.1	17.1
CISI	76	63	7.7	8.6
CACM	52	32	9.6	14.3

*All queries in the Medlars collection are affected by thesaurus class indexing, hence no reduced collection is generated in this case.

This algorithm depends only on the setting of one parameter (*minimum document frequency*). It was applied to the four test collections, using the frequency values shown (which correspond in most cases to the best case values for minimum document frequency) . The results are shown in Table 7.

As Table 7 shows, this approach yields significant improvements in retrieval effectiveness. The results, while substantially less than those produced by the original algorithm, are generated at little cost.

SUMMARY

The performance of an algorithm for automatically constructing a global thesaurus based on the term discrimination value model and a complete link clustering of the document file was examined by applying it to four standard test collections. The results of the experiments indicate that the algorithm can be used to produce useful thesauri, substantially improving retrieval effectiveness in these cases. An attempt to differentiate between the thesaurus classes produced (i.e., between classes which were useful and nonuseful to the retrieval process) by using discrimination value was not successful. An attempt to produce a useful thesaurus by basing membership in a thesaurus class on the discrimination values of the component terms rather than minimum document frequency was also unsuccessful. A revision of the original algorithm, in which low level clusters (all of whose children are documents) are used along with minimum document frequency to form thesaurus classes, is proposed. Results indicate that this approach may be useful in some cases; for two of the four collections, this algorithm produced results comparable to those of the original algorithm at very little cost in terms of time and effort.

ACKNOWLEDGEMENTS

This work was supported in part by the University of Minnesota Graduate School Grant-in-Aid for Research.

Table 7

Improvement Using Thesaurus Generated by the Revised Algorithm (Best Case)

Collection	Frequency	Percentage over Base Case
ADI	15	5.9
Medlars	45	13.6
CISI	30	8.2
CACM	90	3.6

REFERENCES

1. Crouch, C. A cluster-based approach to thesaurus construction. *Proceedings of the Eleventh International Conference on Research and Development in Information Retrieval*; 1988; Grenoble, France.
2. Crouch, C. An approach to the automatic construction of global thesauri. *Information Processing and Management*, 26(5):629-640; 1990.
3. Cleverdon, C.W.; Mills, J.; Keen, M. Factors determining the performance of indexing systems. *Aslib Cranfield Project*, Vol.1; 1966.
4. Salton, G. Scientific reports on information storage and retrieval (ISR 11). Department of Computer Science, Cornell University, 1966.
5. Salton, G. Scientific reports on information storage and retrieval (ISR 13). Department of Computer Science, Cornell University, 1968.
6. Salton, G., ed. *The SMART retrieval system--experiments in automatic document processing.* Englewood Cliffs, N.J.: Prentice-Hall; 1971.
7. Sparck Jones, K. *Automatic keyword classification for information retrieval.* London: Butterworths; 1971.
8. Fox, E. Lexical relations: enhancing effectiveness of information retrieval systems. *ACM SIGIR Forum*, 15(3):5-36; Winter 1980.
9. Wang, Y.; Vandendorpe, J.; Evens, M. Relational thesauri in information retrieval. *Journal of the American Society for Information Science*, 36(1):15-27; 1985.
10. Fox, E.; Nutter, J.; Ahlswede, T.; Markowitz, J. Building a large thesaurus for information retrieval. *Proceedings of the Second Conference on Applied Natural Language Processing*; 1988; Austin, TX.
11. Fox, E. Building the CODER lexicon: the Collins English dictionary and its adverb definitions. Tech. Report 86-23, Department of Computer Science, VPI&SU, Blacksburg, VA.
12. Chen, H.; Lynch, K. Semantics-based information management and retrieval: A knowledge discovery approach. IEEE Transactions on Systems, Man and Cybernetics, 1992.
13. Chen, H.; Dhar, V. Cognitive process as a basis for intelligent retrieval systems design. *Information Processing and Management*, 27(5): 405-432; 1991.
14. Salton, G.; Yang, C.S.; Yu, C. T. A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science*, 26(1):33-44; 1975.15.
15. Salton, G.; McGill, M. *Introduction to modern information retrieval.* New York: McGraw-Hill; 1983.
16. El-Hamdouchi, A.; Willett, P. An improved algorithm for the calculation of exact term discrimination values. *Information Processing and Management*, 24(1):17-22; 1988.
17. Voorhees, E. The effectiveness and efficiency of agglomerative hierarchic clustering in document retrieval. Ph.D. Thesis, Department of Computer Science, Cornell University, 1985.
18. Voorhees, E. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. Tech. Report 86-765, Department of Computer Science, Cornell University.
19. van Rijsbergen, C.J. *Information retrieval*, 2nd edition. London: Butterworths; 1979.
20. Buckley, C. Implementation of the SMART information retrieval system. Tech. Report 85-686, Department of Computer Science, Cornell University.

21. Fox, E. Characteristics of two new experimental collections in computer and information science containing textual and bibliographic concepts. Tech. Report 83-561, Department of Computer Science, Cornell University.

22. Fox, E. Extending the boolean and vector space models of information retrieval with p-norm queries and multiple concept types. Ph.D. Thesis, Department of Computer Science, Cornell University, 1983.