

A COMPUTER SCIENCE VIEW OBTAINED
BY AUTOMATIC DOCUMENT PROCESSING

G. Salton and D. Bergmark

TR 77-309

Computer Science Dept.
Cornell University
Ithaca, N.U. 14853

A Computer Science View Obtained by Automatic
Document Processing

G. Salton and D. Bergmark*

Abstract

Automatic document classification techniques have been widely advocated for the study of various fields of learning, the identification of individual research topics and of influential contributors in a given field, and for information storage and retrieval purposes.

Several thousand research documents in computer science are automatically classified in the present study, leading to the generation of a taxonomy which reflects the state of computer science as of 1974. The popularity of various subject areas in the field is assessed, and the clustering characteristics of particular document classes and authors is given.

1. Clustered Files

A clustered file is a structure in which the stored records are grouped into classes, or clusters of records according to affinities that may exist between them. Typically each file item, or record may be characterized by a set of properties, or attributes reflecting the contents of the record. A similarity coefficient between pairs of records may then be computed as a function of the number of common attributes between them, or alternatively as a function of the weights

* Computer Science Department, Cornell University, Ithaca, New York, 14853.

of the common attributes. Whenever the similarity coefficient between items exceeds a given threshold the tendency is to assign them to a common cluster.

It is convenient to identify each cluster by a set of cluster attributes consisting of a combination of the attribute sets for the individual items in that class. The set, or vector, of cluster attributes is known as the class profile, or centroid vector. A rapid file search and retrieval process suggests itself based on a comparison of the information requests with some of the centroid and record vectors: initially, the requests are compared against the centroids only; this initial query-centroid comparison is then followed by a selective query-record comparison for the records only whose centroids are known to be similar to the queries. If too many clusters are defined in a given environment, it may be useful to group the corresponding cluster centroids into superclusters, identified by supercentroids. The latter, in turn, may be grouped into still larger aggregates until eventually a few large subsections of the file are left to constitute the complete file.

Such an arrangement produces a hierarchical search system in which the search requests are first compared against the identifiers attached to the largest subsections of the file; this comparison in turn gives access to smaller clusters included in the larger ones; by proceeding down into the cluster hierarchy one is able eventually to reach the individual records included in some of the lowest-level clusters. [1,2]

The main problem with the practical use of clustered files arises not in connection with file searching but rather with the initial cluster generation. The best known clustering methods are all based on the

availability of complete similarity information specifying the similarity between pairs of file items. When the file is large the computation of the similarity matrix (of order n^2 for n items) becomes too expensive to be carried out in practice. In recent years, cheaper clustering methods have been proposed which are theoretically less appealing because the final cluster structure depends on the order in which the file items enter into the clustering process; these inexpensive procedures do however generate, and update the file in a time proportional to $n \log n$ for n items. [1,2] One may hope that clustered files may thus be rendered practical for many applications.

File clustering methodologies are used in many fields of learning. When the objects to be classified are written documents or books, file clustering techniques based on bibliographic citation data have been used to construct networks of paper [3], to evaluate the significance of individual research papers or topics [4,5,6], to predict the importance and influence of research workers [7], to assign content identifiers to bibliographic data [8,9,10], to evaluate the importance of certain journals in the literature [11], and to control file search and retrieval procedures.

In the remainder of this study a clustered collection of 3,520 documents in the computer science literature is examined and conclusions are drawn regarding the scope of the field, the importance of various subareas, and the characteristics of certain documents and authors:

2. Automatic Clustering of Computer Science Documents

A computer science collection was initially generated by taking 334 base articles published in the ACM literature (ACM Communications, ACM Journal, ACM Computing Surveys) during 1974, or listed in Computer and Control Abstracts during 1968 and 1971. All the references included in the bibliographies of the base articles were added to the original 334 items to form a complete collection of 3,520 distinct articles. Ideally a document collection might be clustered by comparing keywords or content identifiers assigned to the various items. In the present case, keywords were not available for most items; attribute vectors were therefore constructed by using bibliographic references and citations. For each paper, the references are defined as the items included in the bibliography attached to that paper; the citations, on the other hand, are references made by outside documents to each given paper. For each paper, an attribute vector can then be defined as

$$D = (r_1, r_2, \dots, r_n, c_1, c_2, \dots, c_m)$$

where r_i is an identifier representing the i th reference, and c_j the j th citation for document D .

Citation vectors such as the one previously given make it possible to group pairs of documents which share many references and/or many citations. Since a direct link between two documents (A refers to B, or B cites A) also provides information that can be used in the clustering process, a self-reference indicator is added to each document vector

to about 270 documents for the P4 clusters.

The cluster tree obtained for the computer science collection is presented in Fig. 2. Each path in the tree from bottom (level 0) to top (level 4) represents the clustering characteristics for a certain set of documents. Thus the left-most path, designated as (P1 P2 P3 P4) represents 1125 items that are first entered into P1 clusters; these P1 clusters are then merged into P2 clusters, which themselves are transformed into P3 and eventually into P4 clusters. The 1125 items described by the left-most path in Fig. 2 appear to fit easily into the hierarchical clustering schema, since they could be successively entered into larger and larger cluster structures. On the other hand, the right-most path in Fig. 2 identifies 12 documents that could not be incorporated into any cluster after four clustering passes. These documents remain "loose" at the end of the process.

The document sets of Fig. 2 are separated somewhat arbitrarily into three categories, identified as "easy", "medium", and "hard" depending on the ease with which they adapt themselves to the clustering process. The clustering paths corresponding to the three categories are shown in the figure together with the document percentage represented by each path. It may be seen that approximately 50 percent of the items are easy to cluster; 20 percent have medium clustering characteristics, and the remaining 30 percent are hard to cluster. The document properties which render an item easy or hard to cluster are examined in more detail in the next section. For the moment it may suffice to note that a surprisingly large number of items (almost 1000 out of 3,500) are difficult to incorporate into the

A multilevel clustering process was used to classify the computer science collection. The process is bottom-up starting with the individual documents and creating increasingly larger clusters on successive (higher) levels of the cluster hierarchy. In pass 1 (P1), all articles that were referred to by one other document only and that themselves did not cite other items in the collection were incorporated into a common cluster with the citing item. This produced 327 pass-1 clusters. The remaining pass-2, pass-3, and pass-4 clusters (P2, P3, and P4) were formed by vector matching operations as previously explained. As one proceeds upward in the cluster hierarchy, the cluster size increases and the amount of similarity (the number of matching citation links) between an item and its peers in the same cluster decreases. The clustering statistics of Table 1 show that cluster size increases from an average of about 9 documents per P1 cluster

found for the 3,520 articles. 3,893 distinct links (either references or citations between items) were included among the references but did no citing of their own. A total of among the references cited other base articles, and 33 base articles were themselves also cite other base articles. 33 base articles not included 13 of the base articles were referred to by other base articles and did plus 3,186 references not previously included among the base articles. In Fig. 1. It may be seen that the collection consists of 334 base articles The make-up of the computer science collection is shown schematically

citations between documents. can then be based on common references, common citations and direct A to provide an appropriate term match. The document grouping process identifier in vector B will then match with the A identifier included in identifying the document itself. When item B cites item A, the A

clustering process. Over 100 documents do not fit into any P1 or P2 cluster, and an additional 830 items are incorporated into P1 clusters that cannot be merged further until at least level 4 of the tree.

The unbalanced tree of Fig. 2 is difficult to utilize for search purposes because an incoming query must apparently be compared initially with 9 level 4 centroids, 2 level 3 centroids, 8 level 2 centroids, 91 level 1 centroids, and finally 12 loose documents at level 0. Dummy nodes, and hence dummy centroid vectors may however be introduced, represented by heavy dots in Fig. 2. The use of dummy vectors increases the storage overhead but simplifies the search: for the structure of Fig. 2, each query must effectively be compared with 13 level 4 centroids only, of which 4 are dummy vectors; these in turn give access to 28 level 3 centroids including 6 dummies, and so on down to the 3520 documents on level 0 of the tree.

The actual clustering results for computer science are examined in the remaining sections of this study.

3. Computer Science Taxonomy

Before building a taxonomy for a given field, it is important to define the elements to be included in the structure. In the present case, the individual documents do not by themselves constitute complete topic classes; on the other hand, the pass-3 and pass-4 clusters, each comprising an average of nearly 100 and 300 items respectively, appear too large for the representation of homogeneous subject areas. The taxonomy is therefore defined in terms of pass-2 clusters, and those pass-1 clusters that are not themselves incorporated in any pass-2 cluster. The corresponding taxonomy is reproduced as Fig. 3.

Each P1 or P2 cluster included in Fig. 3 is listed under one or more topic headings suggested by the cluster contents. The overall structure follows the model used to classify articles in the Encyclopedia of Computer Science. Six main headings are used including mathematics for computing, hardware, theory, software, computer systems, and methodology. [12] Two additional headings are included although the subjects are not well represented by the collection under study: applications and societal aspects, respectively.

The following coding system is used in Fig. 3 to identify the various research clusters:

- a) three digit identifiers starting with a 6 are used for the 41 P2 clusters that appear in the easy clustering category;
- b) four digit identifiers starting with a 6 represent the 20 P2 clusters with medium clustering characteristics;
- c) four digits starting with digits 1, 2, 3, 4 or 5 represent the 57 P1 clusters with medium clustering properties;
- d) the remaining P1 clusters which are hard to cluster are identified by 5 digit codes starting with digits 1 to 5 and ending in a 0.

It may be noticed that a number of clusters particularly in the easy clustering category are listed several times in the chart, the various occurrences being identified by subscripts. These clusters generally represent popular topics that cannot easily be identified by a single description. Thus, cluster 641 is described by "search systems" and "hashing" under Information Storage and Retrieval (Methodology), by "trees" and "decision trees" under Program and Data Structures (Software), and finally by "access methods"

The taxonomy of Fig. 3 may be further examined from three viewpoints: active vs. inactive topic classes; subjects that lend themselves to the formation of homogeneous research clusters, and others that do not; and finally, topics that relate well to other taxonomic areas, and others that remain isolated.

A number of topic classes proved surprisingly active in the computer science literature. Among them are

- a) Mathematical computations (random number generation, contingency tables, etc.) under Mathematics for Computing;
- b) Storage management (segmentation, partitioning, virtual systems) under Computer Systems;
- c) Coding systems (error correcting codes, convolution codes, etc.) under Computer Systems;
- d) Pattern recognition (line drawings, curve detection feature analysis) under Computer Systems;
- e) Computational models (mathematical programming, simulation, decision theory, etc.) under Methodology;
- f) Combinatorial mathematics (game theory, classification, flow and network theory, scheduling theory, etc.) under Methodology; and
- g) Graph theory (shortest path, connectivity, planarity, etc.) listed under combinatorial mathematics in the Methodology section.

On the other hand, certain other topic classes are poorly represented by the literature sample under study. Among them are not surprisingly Applications, and Societal Aspects, these topics being poorly covered in the ACM literature at large. Additional topics, such as data base management, were largely absent

All of these topics are covered quite extensively by the literature sample under study, and a case can be made for breaking up these subjects and assigning separate main headings. The point of view taken here asserts that from a computer science viewpoint the techniques and methodologies developed for the automatic treatment of these topics is of principal interest rather than the particular application. The Methodology section must then be distinguished from the various applications areas (computers in business, computers in education, etc.), where the papers specifically appeal to readers that are not primarily computer scientists.

queueing theory.
mathematical programming, simulation, decision theory, information theory, and
and network theory; and mathematical or computational models such as mathematical; a number of subjects tangential to computing such as graph theory
includes a variety of techniques, such as pattern matching and combinatorial
storage and retrieval, and data base systems. In addition, the section also
computer science topics such as artificial intelligence, graphics, information
appears to be somewhat of a catch-all, since it encompasses a variety of
is not used in the Encyclopedia of Computer Science. The methodology section
a few words about the main heading labelled Methodology, which incidentally
Before examining the taxonomy in more detail, it may be useful to say
These documents are discussed separately in the next section.

57 hard to cluster single items that did not fit into any P1 or P2 cluster.
The only documents not taken into account in the taxonomy of Fig. 3 are
"moment method" (Pattern Recognition under Methodology).
as "splines" (Approximation Theory under Mathematics for Computing) and
under Storage Management (Computer Systems). Similarly, cluster 610 appears

from the literature of 1974, and are therefore not represented by the research clusters of Fig. 3.

Another important characteristic of the taxonomy is the ease with which homogeneous research clusters are produced for certain subject areas, but not for some others. The most representative clusters from the point of view of subject identification are the P2 clusters containing an average of about 30 documents each. It turns out that the P2 clusters which are directly related to recognizable subtopics are unevenly distributed among the various subject headings. A number of areas are listed in Table 2 according to the ease with which they lend themselves to the construction of P2 clusters.

It may be seen from the Table that the topic classes generally recognized as the core of computer science fit easily into the structure; they are represented by P2 classes and exhibit many relations with other subject areas. Among these is the complete Software section, including programming and programming languages, and the systems topics (operating systems, etc.) included under Computer Systems. Also listed among the "easy" clustering topics is the section on Mathematics for Computing and the various Methodology sections that were mentioned earlier.

The topics that fit less well into the taxonomy are Hardware and Applications, and a number of theoretical topics such as automata theory, formal languages, and theorem proving that generate many small P1 clusters, but are difficult to merge into larger substructures.

Consider now the actual relationships that exist between the various subjects included in the classification of Fig. 3. The following topic similarities are easily recognized by examining the research clusters assigned

to more than one area:

a) Computer Systems vs. Hardware

storage management - CPU logic design
 access methods, scheduling - storage allocation
 microprogramming - terminal equipment
 networks - computers

b) Software vs. Theory

software engineering - program correctness
 program linguistics - formal languages

c) Methodology (Pattern Matching) vs. Mathematics for Computing

pattern analysis - splines
 character recognition - approximation theory
 iterative analysis - matrix computation

d) Computer Systems vs. Methodology

storage management - queuing theory
 networks - flow analysis.

The existence of these relationships is easy to verify. In each case, the boundary line between topics is difficult to draw, for example between Systems and Hardware, or between Software and Theory; and the connection between mathematical methodology and the Mathematics section is also obvious. Indeed the Methodology section relates directly to most other portions of the taxonomy.

A general summary of the taxonomy is not easy to draw: the topics of computer science are all represented and recognizable in the classification. The wide open Methodology section which impinges on so many related topic classes demonstrates, however, that the boundaries of computer science are not well-defined. This is, of course, confirmed by the fact that computing is a

Table 4(b) contains surveys and historical articles where the situation is the dual of the one previously described: instead of heterogeneous citations, there are many heterogeneous references to other items. Tables 4(c) and 4(d) contain programming language descriptions and various interdisciplinary items that are in part tangential to computing. Here both the references as well as the citations may relate to several different topic areas.

The last class of items in Table 4 covers well-known research articles. Several different reasons may be given in this case for the clustering failure. Some of the articles are seminal works starting new endeavors in the field for which citations had not accumulated by 1976; some of the other papers contain interdisciplinary references; and a few finally exhibit no references at all.

One may conclude from the foregoing exposition that the clustering characteristics of a given item have nothing to do with the importance of the item in the field. Completely novel items that open up new areas are difficult to classify, as are basic reference works and interdisciplinary items. In these circumstances it is difficult to predict in advance where a given item, or a given author, will fit into the cluster organization for a given field.

This is confirmed by examining the list of 25 "prolific" authors shown in Table 4. These authors are represented at least 10 times in the data base of 3520 articles; they are entered in the table under the "easy", "medium", or "hard" label using the criteria of Fig. 2. That is, if a substantial number of articles by a given author follow the clustering paths described as "easy" in Fig. 2, the author is listed in the "easy clustering" category. The same applies to the medium and hard categories. Once again, it is clear that clustering ease does not seem to be correlated with eminence in the field.

popular topic in many areas of the sciences and humanities in addition to computer science itself.

4. Authors and Papers

It is not possible in the present context to give a detailed analysis of

individual documents and authors. It may, however, be of interest to inquire

briefly into the nature of those documents that did not fit into any of the

P1 and P2 clusters [paths (---), (---P3-), and (---P4)] of Fig. 2].

In so doing, one must remember that an item is incorporated into a cluster

whenever its similarity with a given cluster centroid is substantially higher

than its similarity with the other centroids. Items that present clustering

difficulties are therefore those that exhibit no substantial similarities with

any of the centroids, or those on the contrary that present affinities with a

large number of centroids. In the latter case, the choice is between entering

the items into all of these clusters simultaneously, or alternatively to keep

them unclustered. The former possibility produces a great deal of storage

overhead, and should therefore be avoided whenever possible.

A list of representative unclustered items is contained in Table 3 arranged

in several different categories. Table 3(a) contains basic reference works

that are cited by a large class of heterogeneous items. Included in this class

are the famous volumes by Knuth on "The Art of Programming". A clustering

diagram for a typical basic reference work is shown in Fig. 4, indicating cita-

tions originating in many different clusters. The item illustrated in Fig. 4

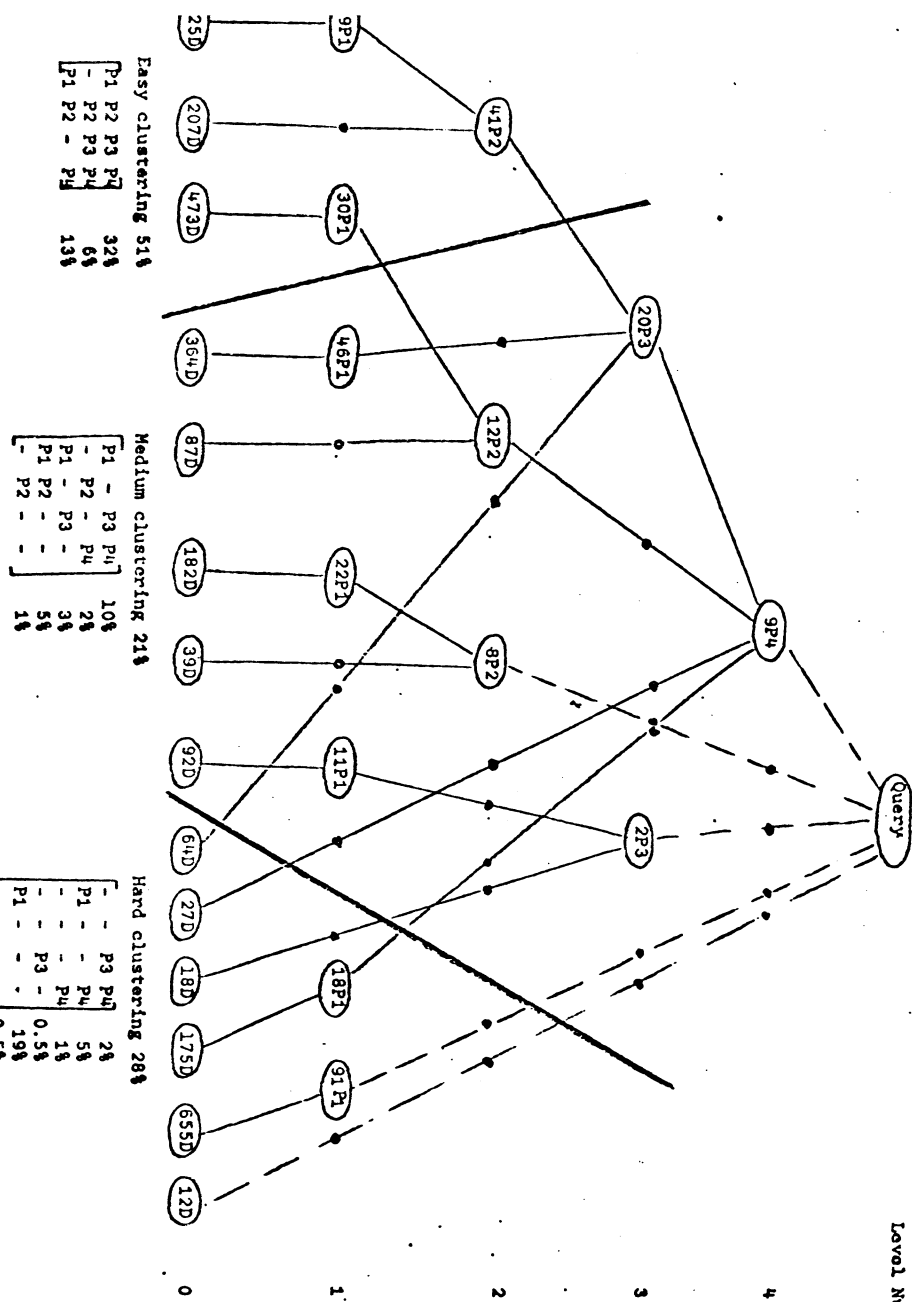
is eventually included in a cluster at level 4 of the cluster tree.

It does appear, however, that authors in the software area are generally included in the "easy" class, whereas many theoreticians exhibit "medium" or "hard" clustering characteristics.

In principle, it seems desirable to draw some overall conclusions about computer science as a scientific discipline, or at least about the state of computer science around 1974. However such conclusions might be shallow and ephemeral. It seems safest then to let the automatic taxonomy speak for itself. In a few years, the exercise might be repeated, possibly with a larger cross-section of the literature to determine whether the core has remained identical and to examine the extent to which the field has evolved.

REFERENCES

- (1) G. Salton, D. Bergmark, and A. Wong, Generation and Search of Clustered Files, Computer Science Technical Report 77-299, Cornell University, Ithaca, New York, January 1977.
- (2) G. Salton, Dynamic Information and Library Processing, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1975, Chapter 8.
- (3) D. J. deSolla Price, Networks of Scientific Papers, Science, Vol. 149, No. 3683, 1965, p. 510-515.
- (4) J. H. Westbrook, Identifying Significant Research, Science, Vol. 132, No. 3435, 1960, p. 1229-1234.
- (5) J. Margolis, Citation Indexing and Evaluation of Scientific Papers, Science, Vol. 155, No. 3767, 1967, p. 1213-1219.
- (6) M. M. Kessler, Bibliographic Coupling Between Scientific Papers, Am. Documentation, Vol. 14, 1963, p. 10-25.
- (7) E. Garfield, Citation Indexing, Historio Bibliography and the Sociology of Science, Third Int. Congress of Medical Librarianship, Amsterdam, May 1969, p. 187-206.
- (8) E. Garfield, Science Citation Index - A New Dimension in Indexing, Science, Vol. 144, No. 3619, May 1964, p. 649-654.
- (9) M. M. Kessler, Comparison of Results of Bibliographic Coupling and Analytic Subject Indexing, Am. Documentation, Vol. 15, No. 3, 1965, p. 223-233.
- (10) G. Salton, Automatic Indexing Using Bibliographic Citations, Journal of Documentation, Vol. 27, No. 2, June 1971, p. 98-110.
- (11) E. Garfield, Citation Analysis as a Tool in Journal Evaluation, Science, Vol. 178, No. 4060, November 1972, p. 471-479.
- (12) A. Ralston and C. L. Meek, Encyclopedia of Computer Science, Petrocelli Charter, New York, 1976.



Easy clustering 51%

P1	P2	P3	P4	32%
-	-	-	-	6%
P1	P2	-	-	13%

Medium clustering 21%

P1	-	P3	P4	10%
-	P2	-	-	2%
P1	-	P3	-	3%
-	P2	-	-	5%
P1	P2	-	-	1%

Hard clustering 28%

-	-	P3	P4	2%
P1	-	-	-	5%
-	-	-	-	1%
-	-	P3	-	0.5%
P1	-	-	-	1.9%
-	-	-	-	0.5%

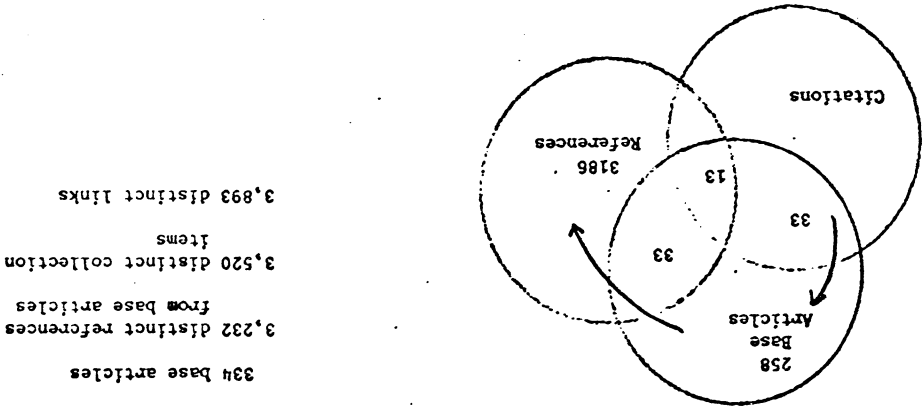
Table 1

Clustering Statistics

3520 original documents	1.0 document/cluster
327 pass-1 clusters	9.5 document/cluster
61 pass-2 clusters	33.0 document/cluster
22 pass-3 clusters	95.0 document/cluster
9 pass-4 clusters	274.3 documents/cluster

Fig. 1

Make-up of Computer Science Collection
 (1974 ACM literature plus 1968 and 1971 articles from
 Computer and Control Abstracts)



division	618	Finite state automata	3396
Approximation Theory	617 ₁ 651 ₂ 21270	pushdown automata	3396
Interpolation	23660	sequential machines	3335 3666
splines	618 ₁ 6020 25130	maximal automata	3418
least squares	603 ₁	stochastic automata	3800 3830 37570
Chebyshev approximation	603 ₁	Formal Languages	648 ₂ 656 ₃ 67550
partial fractions	3829	context free grammar	1010
Ordinary Differential Equations		parsing	1664 ₂
Initial value problems	6000	right linear grammar	3750
boundary value problems	6076	closure preservation	3539
Partial Differential Equations	640 ₂	ambiguity	37670 ₂
Numerical Integration	616 ₂ 21950	production systems	27710 25800
Newton Cotes	3790	Theorem Proving	645 647 ₁
Fast Fourier Transform	619 621 ₂ 3760	undecidability	13600
cyclic convolution	3773	refutation	13600
signal theory	35670 37300	mathematical logic	32500 ₁
Matrix Computations	615 ₁ 625 ₁ 3900	Theory of Algorithms	600 ₂ 630 ₃ 650 ₃
eigenvalues	619 ₃	Complexity, Computability	607 ₃ 6057
root finding	640 ₁	Unsolvability	
clustering	620 ₁	Program Corrections	630 ₂ 643 ₁ 656 ₁ 75
determinants	2765	loop predicates	2572
Error Analysis, Stability	623 ₂ 651 ₃	Program Optimization	642 ₂ 650 ₂ 10000
Statistical Computations	620 ₂	Boolean Algebra	629 ₁ 12550
Kolmogorov Shirkov distr.	2931	Software	
random number generation	2932 6033 ₂	Programming	
contingency tables	22000	structured	601 ₁ 620 ₁ 629 ₁ 630
classification error	28732	debugging	32670
digital filtering	3737	practice	629 ₁ 62710
Hardware		concurrent	22500
Performance reliability	6360	style	620 ₂
Computers		System Software	
super computers	636 ₂	opcode interpretation	27320
microprogrammed	3784 ₁	opcode utilization	37670
Computer Design		self stabilizing systems	16670
virtual machine design	37530	compilers	630 ₂
control structures	40510	Programming Languages	
parallel microcomputer	3785 ₁	Pascal	641 ₂
Binary Systems		high-level	6300
failure tolerant design	3797	Cobol evaluation	51100
storage allocation	600 ₃	Simula and Fortran	32200
Peripheral Equipment		AI languages	1550 ₁
terminals	672 ₂	Math. logic languages	32500 ₂
Central Processing Unit		Software Engineering	630 ₂
logic design	650 ₂	portability	
		chief programmer team	620 ₂
		Program and Data Structure	632 5810
		decision tables	640 ₁
		strings	630 ₁
		trees, decision trees	641 ₂ 640 ₂
		stacks	
		Programming Linguistics	643 ₂
		state transition techniques	3221
		parser	640 ₁ 640 ₂ 670
		string processing	630 ₂

Computer Science Taxonomy
Fig. 2

error correcting codes 2120 2030 2120
convolution codes 2503
burst errors 470a
radix transformation 618 3768

Storage Management

extendible arrays 12540
dynamic allocation 11390
register allocation 1664 5253
paging 600, 607, 603, 653, 642, 7500 7502
partitioning 6189
virtual system 27500
file organization 607
storage structures 654
access methods 600, 601
segmentation 637

Time Sharing

scheduling methods

Networks

computer utility

Processing Mode

parallel processing 607, 636, 658
multiprocessing 627, 653, 654, 2785

Microprogramming

612, 622, 3700

System Management

security 627, 627
cryptography 627
protection 627
data compression 22520

Applications

Basic

16450

Social Science Applications

77010

Computing in Business

75060

Education

75070

trackline machines 75080
time table construction 625

Societal Aspects

privacy 627
human systems 627

Gen	P2 clusters easy clustering
Gen to Sum	P1 clusters median clustering
Gen	P2 clusters median clustering
Gen to Sum	P1 clusters hard clustering

expert method 610
line drawings 145
image processing 617, 3876
character recognition 617, 620, 642
stereic analysis 612
bubble cluster scan 6811
curve detection 2762
chromosome recognition 24070
mechanical diagnostics 2015

Artificial Intelligence

heuristic processing 647
speech recognition 166, 2003
fact retrieval
AI languages 1552

Computational Models

mathematical programming 605, 651, 6052
integer programming 11410
dynamic programming 608
nonlinear algebra 48510
simulation
decision theory 627
information theory 24510
queueing theory 652, 11360

Combinatorial Mathematics

matching theory 620
game theory 35410
classification 620, 660, 2022
flow analysis 605, 607, 655
scheduling theory 23610, 37620
network theory
routing network 610, 29240
graph theory
graph coloring 627, 625
connectivity 1245, 37510
partitioning 626
chromatic number 640
shortest path 2285, 4827, 6052, 24970
Hamiltonian path 12460, 24400
spanning tree 2120, 3730
graph matching 4255
planarity 1244
transitivity sets 22620

Information Storage and Retrieval

29260
automatic indexing 48600
search systems 640, 641, 16420, 20200
document retrieval 16700, 27000
retrieval evaluation 26160
routing 603

Data Base Systems

Computational Linguistics
content analysis 3100

Symbol Manipulating Systems

algebraic computations

Graphics

hidden surface 612
picture display 620, 20200
scanning
half tone 622

Table 2

Homogeneity of Clusters of Topic Classes in Taxonomy

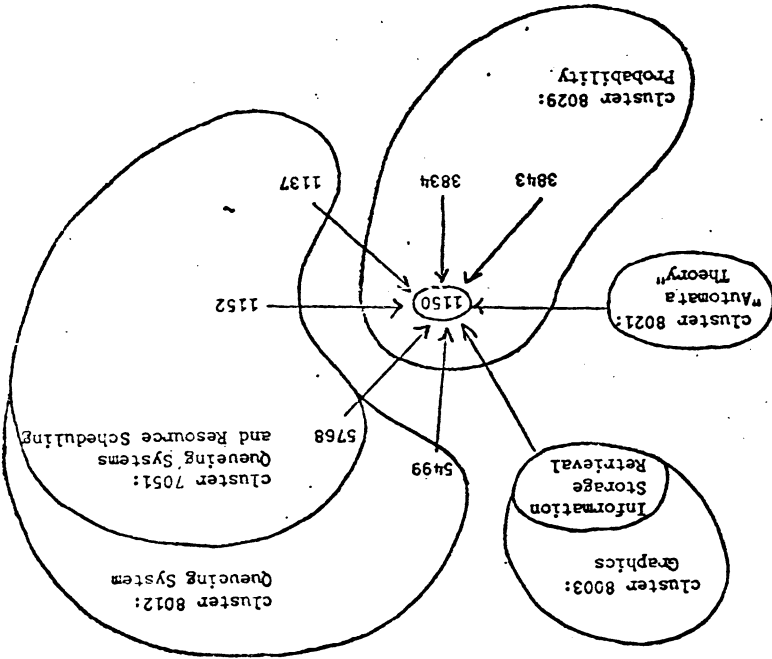
<p>Homogeneous Research Areas Represented by P2 clusters</p>	<p>Nonhomogeneous Areas Not Represented by P2 clusters</p>
<p>Software (complete section) Mathematics for Compiling (except for statistics) Computer Systems Storage Management Processing Modes Systems Management Methodology Pattern Recognition Computational Models Combinatorial Math.</p>	<p>Hardware Composite Design Theory Automata Theory Formal Languages Theorem Proving Computer Systems Coding Systems Applications (complete section) Methodology Information Storage and Retrieval</p>

Fig. 4

(Feller, An Introduction to Probability Theory)
Clustering Diagram for a Basic Reference Work

A
B
A refers to B
B is cited by A

8000 Level 4 clusters
7000 Level 3 clusters



Ginsburg	An Introduction to Mathematical Machine Theory
Gries	Compiler Construction for Digital Computers
Harrison	Introduction to Switching and Automata Theory
Härtmanis and Stearns	Algebraic Structure Theory of Sequential Machines
Husson	Microprogramming Principles and Practice
Knuth	The Art of Programming, Volumes 1 and 3
Nilsson	Problem Solving Methods in Artificial Intelligence
Rabin	Probabilistic Automata

a) Basic Reference Works

Bobrow and Raphael	Comparison of List Processing Computer Languages
Dreyfus	Shortest Path Algorithms
Feldman and Gries	Translator Writing Systems
Geoffrion and Marston	Integer Programming: A Framework and State-of-the-Art Survey
London	Bibliography on Proving Program Correctness
Martin	Sorting

b) Survey and Tutorial Articles
(many references)

Falkoff and Iverson	APL/360 - User's Manual
Naur	Revised Report on the Algorithmic Language Algol 60
Van Wijngaarden	Report on the Algorithmic Language Algol 68

c) Programming Language Descriptions

Sample Items with Hard Clustering Characteristics

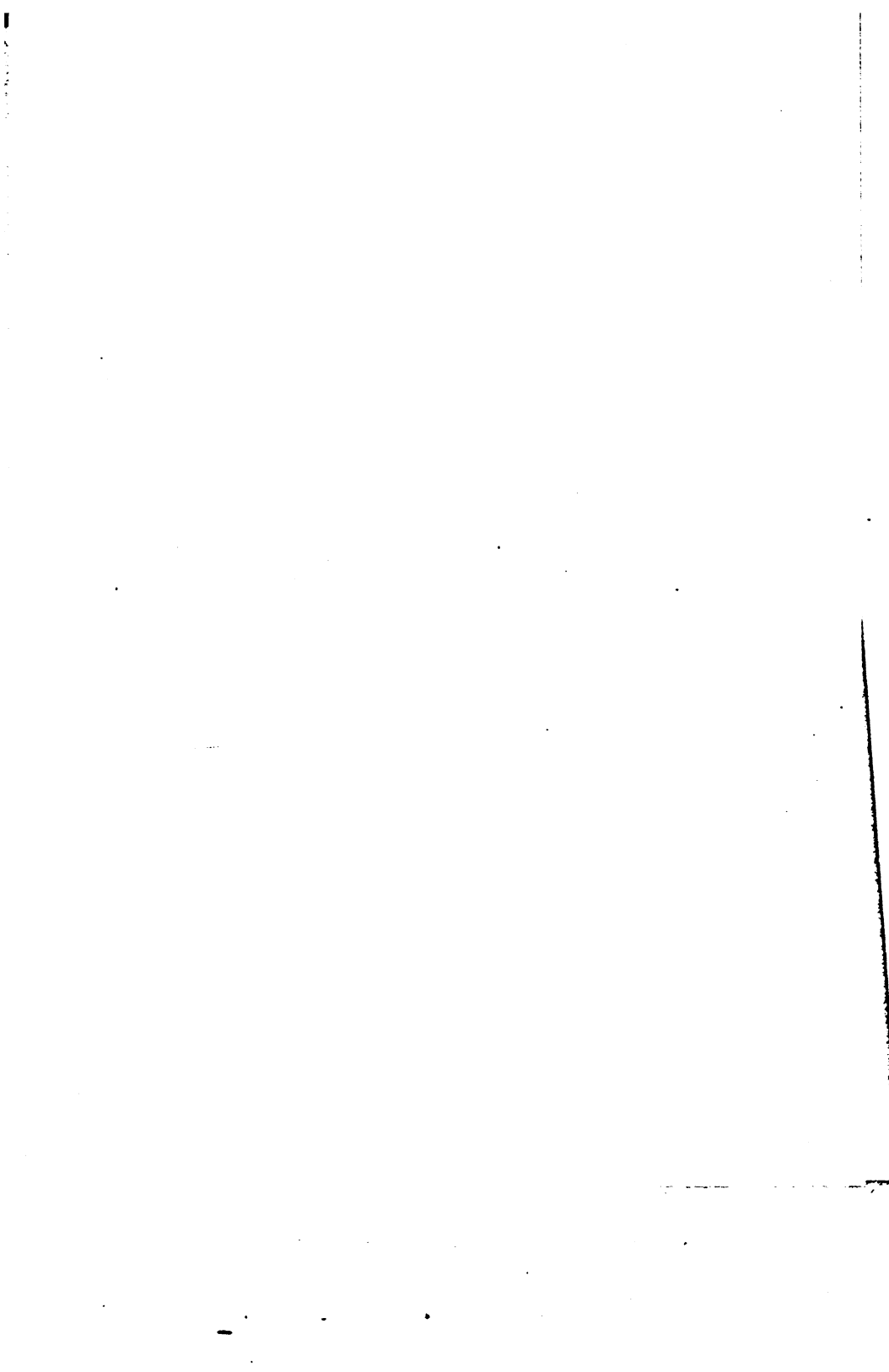
Berlekamp	Algebraic Coding Theory
Berge	Theory of Graphs and its Applications
Birkhoff and MacLane	Survey of Modern Algebra
Chomsky	Aspects of the Theory of Syntax
Peterson	Error Correcting Codes
Salton	Automatic Information Organization and Retrieval
Shannon and Weaver	Mathematical Theory of Communications
Weinberg	Psychology of Computer Programs

Tangential or Interdisciplinary Items
(references to and citations from several areas)

Cline	Curve Fitting Using Splines Under Tension
Dahl, Dijkstra, and Hoare	Structured Programming
Dijkstra	Structures of the THE Multiprogramming System
Floyd	Assigning Meaning to Programs
Greibach	Hardest Context Free Language
Hopcroft and Tarjan	Efficient Algorithms for Graph Manipulation
Huffman	A Method for the Construction of Minimum Redundancy Codes
Karp	Reducibility among Combinatorial Problems
Knuth	Empirical Study of Fortran Programs
Landin	Correspondence between Algol 60 and Church's Lambda Notation
McCarthy	A Basis for a Mathematical Theory of Computation
McNaughton and Anderson	Regular Expressions and Static Graphs for Automata
Nievergelt and Wong	Dynamic Storage Allocation for Binary Search Trees
Ross	The AED Approach to Generalized Computer Coded Design

Research Articles
(Starting new areas, no references,
interdisciplinary references)

Sample Items with Hard Clustering Characteristics



Number of Pages
 With Clustering Characteristics

Author Name	Total Pages for Author	Easy	Medium	Hard
-------------	------------------------	------	--------	------

1. A. V. Aho	17	16	2	1
2. E. G. Coffman	21	16	2	3
3. P. J. Denning	11	10	0	1
4. H. Freeman	14	14	0	0
5. H. S. Fu	11	11	0	0
6. I. J. Good	11	10	4	1
7. M. A. Harrison	12	11	1	1
8. C. A. R. Hoare	13	13	0	0
9. S. V. Pollack	16	17	0	2
10. C. V. Ramamoorthy	19	17	3	3
11. J. D. Ullman	34	28	3	3
12. N. Wirth	11	11	0	0
13. C. W. Wong	12	10	0	2

a) Authors with Easy Clustering Characteristics

14. E. W. Dijkstra	20	13	2	5
15. J. Hartmanis	10	4	4	2
16. IBM	20	11	3	6
17. D. E. Knuth	23	14	3	6
18. A. Rosenfeld	19	8	10	1
19. J. R. Slagle	11	6	2	3

b) Authors with Medium Clustering Characteristics

20. D. G. Bobrow	12	1	8	3
21. J. Ginsburg	11	2	5	4
22. J. E. Hopcroft	12	2	2	8
23. A. L. Rosenberg	13	0	4	9
24. G. Salton	13	1	1	11
25. R. E. Tarjan	13	3	0	10

c) Authors with Hard Clustering Characteristics

Total for 25 Authors 379 63% 14% 23%

Clustering Characteristics for Prolific Authors
 (at least 10 papers per author)

Table 4