A CITATION STUDY
OF THE
COMPUTER SCIENCE LITERATURE

by

G. Salton*
and
D. Bergmark+

TR79-364

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

+Department of Mathematics
Ithaca College
Ithaca, New York 14850

This study was supported in part by the National Science Foundation under Grant DSI-77-04843.

A CITATION STUDY OF THE COMPUTER SCIENCE LITERATURE

G. Salton* and D. Bergmark+

Abstract

The bibliographic references and citations which exist between documents in a given collection enviroment can be used to study the history and scope of particular subject areas and to assess the importance of individual authors, documents, and journals. A clustering study of the computer science literature is described using bibliographic citations as a clustering criterion, and conclusions are drawn regarding the scope of computer science, and the characteristics of individual documents in the area.

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

+Department of Mathematics, Ithaca College, Ithaca, New York 14850

This study was supported in part by the National Science Foundation under grant DSI-77-04843.

1. Introduction

It is well-known that the development and scope of individual scientific disciplines is determined to some extent by the structure of the corresponding literature. A literature study makes it possible, in particular, to identify certain relationships between the documents of a given collection, to assess the influence of individual authors, to recognize subdisciplines, and to trace the historical development and progress in the area of interest.

The structure of the literature in a given field can be characterized in several different ways, for example, by constructing subject indexes and classification systems and by considering also author and title listings of various kinds. With the advent of computers, it has become possible to process bibliographic information automatically. Citations and references between documents have accordingly been used as ingredients in many large-scale literature studies.* It is possible, in particular to consider the bibliographic references and citations attached to individual items in the literature as alternative expressions of document content. Citation and reference frequency counts can also be used in part to assess the importance of individual documents, or of complete document collections, assuming that the citation frequency reflects to some extent the influence of an item in the field. By extension, the same argument may apply to

*A distinction is made in this study between bibliographic references which comprise the items included in the bibliographies attached to individual documents, and the citations which are references made by outside documents to a given bibliographic item.

individual authors of documents, or to selected author groups. Finally, citations and references can be followed from item to item, and the resulting "network" of papers can serve as a basis for a variety of historical studies and for an examination of the development of individual disciplines.

A citation analysis of computer science documents is described in the present report, and conclusions are reached regarding the scope of the field and the importance and structure of certain documents in the area.

2. Bibliographic References and Citations

Consider a collection D of documents in a given field of study. The documents may be identified and distinguished from each other by assigning content identifiers in the form of terms, keywords, or descriptors to the individual items. A given document D_i can then be represented as a term vector

$$D_i = (d_{i_1}, d_{i_2}, \ldots, d_{i_t}) \qquad (1)$$

where d_{i_j} represents the j^{th} term (or sometimes the weight of the j^{th} term) assigned to item D_i. The term vector representation makes it possible to compare pairs of documents with each other, or alternatively incoming user queries with individual documents, by using a similarity function s to generate similarity coefficients between corresponding pairs of term vectors. Thus, the coefficient $s(D_i, D_j)$ could represent the similarity between items

D_i and D_j, and $s(Q,D_i)$ the similarity between a query Q and an item D_i. Typical vector similarity functions s may be the inner product, or the cosine of the angle between the term vectors.

The term vector representation of a given document is normally determined without considering the total structure of the collection. That is, the individual terms assigned to a given item are often chosen without taking the remainder of the collection into account. The documents do not, however, exist in isolation, but are related to each other in various ways. The references used to form the bibliographies of the various items represent one type of relationship indication. Inverse references, or citations, made to a given item from the outside represent additional relationships between bibliographic items. In principle, it should then be possible to represent each document by a bibliographic attribute vector of the form

$$D_i = (r_{i_1}, r_{i_2}, \ldots, r_{i_n}, c_{i_1}, c_{i_2}, \ldots c_{i_m}) \qquad (2)$$

where r_{ij} is an identifier representing the j^{th} reference, and c_{ik} the k^{th} citation for document D_i.* The previously mentioned vector similarity operations can be performed using term similarities or citation and reference similarities (expressions (1) or (2)), or the two vector forms could be combined by adding the bibliographic information to the normal index terms.

*The references can normally be obtained directly from the bibliographies of the individual items. To find the citations, it is necessary to consult a citation index, or alternatively to process the reference lists using appropriate merge and sorting operations.

A number of studies have been made in which bibliographic indications have replaced the standard terms and keywords for purposes of content identification and document retrieval. [1-4] It is generally found that the bibliographic indicators tend to produce more specific content identifications than normal keywords, and that comparable retrieval results can be obtained for the bibliographic and the term vector representations. In some retrieval environments the bibliographic identifiers may present the obvious advantage of being immediately available whereas the terms and keywords might have to be generated by some more or less controllable process. Even when objective, keyword-type identifiers can be generated, the conventional term vectors might usefully be supplemented by bibliographic references and citations to form extended content representations.

Bibliographic information of the type shown in expression (2) can be used also for purposes other than document indexing. Thus, it may be tempting to add the number of citations (the number of C_{ik} terms in the bibliographic vectors) for individual documents, or collectively for all the documents written by a given author, or for even larger groups of several different authors, and to draw conclusions concerning the importance and influence of the corresponding authors and papers. Citation counts are easily generated for many scientific disciplines by consulting the well-known Science Citation Index. [5,6]

Of course, it is easy to think of reasons indicating that citation counting would prove to be meaningless: the number of citations which a given paper may be expected to attract might depend on extraneous factors such as the author's reputation, the circulation of the journal in which the given paper is published, the particular subject matter at hand, the dissemination of reprints of the article, the coverage of the subject matter by secondary indexes, the self-citation habits of the particular author, and so on. Indeed, the literature is full of stories where papers allegedly containing errors, possibly purposely introduced by the authors, are found to attract unusually large numbers of citations from people interested in rectifying the erroneous information.

Before jumping to conclusions, one must however remember that the act of citing a given document is comparatively rare -- of a random 100 published papers, 40 are not cited at all in any given year, 50 more are cited only once in a year's time, and only 10 are cited more than once. [7] The previously mentioned objections are normally found to produce at most second-order effects, and serious bibliographic studies have repeatedly confirmed that global citation frequencies correlate well with other indicators of merit. [8-11] This does not imply in any way that scientists should be ranked in order of merit according to some raw citation frequency count. It does mean that scientists attracting impressive numbers of citations are likely to be impressive and influential contributors to their field.

This is confirmed, for example, by evidence showing that Nobel prize winners are invariably members of an elite group of scientists comprising the top 0.1 percent of all cited authors. [6] Additional studies point to the use of citation frequency methods to obtain measures of quality of collections of individuals, such as for example, members of advisory commissions or of university science departments. [12]

Such methods may be applicable to the computer science field as shown by the sample data of Table 1 where the total citation frequencies for the years 1976 and 1977 are shown for a number of high-frequency authors consisting of persons having at least ten research papers included in a test collection of several thousand computer science documents.* The data of Table 1 are taken from the Science Citation Index (SCI) and must be appropriately interpreted. (For example, scientists who publish only jointly-authored papers may be at a disadvantage because the SCI contains citation data only for the first author of each group of joint authors; this accounts for the bracketing of two particular authors included in Table 1). Overall, most readers will find the listing of Table 1 neither surprising nor unreasonable.

Citation counts can be applied not only to personal authorship data but also to complete journal and other publication data. Once again the basic idea consists in obtaining a total citation frequency for the issues of a given journal and interpreting the results to assess journal impact and importance. The resulting

*The test collection of 4,231 articles is described later in this study.

data can be used in turn to define core collections of important journals, and to make decisions concerning individual or public journal subscription policies. A so-called "journal impact factor" is normally used to rank the journals in decreasing order of importance, defined as the number of citations attracted by the collection of articles published by the journal in a given period of time, divided by the total number of citable articles for that journal during the time period. [13-16] When the journal impact factor is used for computer science, it is found as expected that the three journals exhibiting the highest impact factors are the IEEE Transactions on Computers, the Journal of the ACM, and the Communication of the ACM. [17,18]

One possibility not so far considered is to use bibliographic references and citations between documents to produce groups of documents with similar citation patterns. This question is considered in the remainder of this report.

3. Citation and Reference Clustering

Classes, or clusters of documents can be used for many purposes, most importantly perhaps to group documents pertaining to common subject areas thereby facilitating search and retrieval activites. Once again the grouping method can be based on similarities between the content identifiers assigned to individual documents, or alternatively on similarities between the citation and reference patterns.

When bibliographic information is used for clustering purposes two possiblities are immediately apparent: Two or more itmes may exhibit similar sets of references; such a relationship is known as bibliographic coupling. [3,19] Alternatively, two or more items may be related by common citation patterns from outside documents; the corresponding factor is known as cocitation similarity. In Figure 1, items A and B are coupled bibliographically and items C and D are cocited. Cocitations are believed to be more significant than bibliographic coupling in pointing to important similarities in subject areas, and for this reason they have been widely used to study the development of individual fields of science, or the structure of science as a whole. [20-22]

To study the make-up of the computer science literature, a sample collection was clustered using bibliographic coupling as well as cocitation similarities. More specifically, the complete bibliographic attribute vectors (expression (2)) for the documents can be compared, and items with sufficiently high vector similarities may be included in common document classes. Since direct links between documents (A refers to B, or B is cited by A) can also provide useful information for the clustering process, a self-reference indicator s_i may be added to the document vector to identify the given document D_i itself. This produces an extended vector

$$D_i = (s_i, r_{i_1}, r_{i_2}, \ldots, r_{i_n}, c_{i_1}, c_{i_2}, \ldots, c_{i_m}). \qquad (3)$$

When item B is cited by item A (see Figure 1(a)), one of the r_{ij} terms included in the A vector will match the self-reference indicator s_i included in the B vector to provide an appropriate term match. The document grouping process can then be based on common references, common citations, and direct citations between documents. [23]

The make-up of the computer science collection used for experimental purposes is shown schematically in Figure 2. The collection consists of 419 base articles chosen from various journals published in 1974, including the IEEE Transactions on Computers, the Journal of the ACM, the Communications of the ACM, the ACM Computing Surveys, and the Proceedings of the IFIP-74 Congress, the latter being added in an attempt to provide some international coverage. The 419 base articles were supplemented by the 3,812 additional articles listed as references of the original 419. It will be seen from Figure 2 that 14 of the base articles were referred to by other base articles and did themselves also cite other base articles. 35 base articles not included among the references cited other base articles, and 32 base articles were included among the references but did no citing of their own. A total of 4,764 distinct links (either references or citations between items) were found for the 4,231 articles.

A multilevel clustering process was used to classify this 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 394 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 magnitude of the similarity (the number of matching citation links) between an item and its peers in the same cluster decreases. The clustering statistics of Table 2 show that cluster size increases from an average of about 9 documents per P1 cluster to about 335 documents for the P4 clusters.

The cluster tree obtained for the computer science collection is presented in Figure 3. 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 1,354 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 1,354 items described by the left-most path in Figure 3 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 Figure 3 identifies 18 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 Figure 3 are separated somewhat arbitrarily into two categories: on the left side of the diagram are 2,203 items, or 52 percent of the total, that are eventually included in some P2-clusters on level 2 of the tree; on the right side are the remaining 2,028 items, or 48 percent of the total, whose clustering characteristics are weak enough to prevent them from being incorporated into any P2-clusters. The P2-clusters consisting of about 33 documents per cluster may be especially representative of well-defined subareas of the field, in contrast to the smaller P1 clusters with only about 9 items per cluster, and the larger P3 clusters with about 95 documents per cluster. Furthermore, the items included on the left side of Figure 3 are located in dense areas of the collection space in view of the ease with which they fit into the cluster structure. These items may then constitute a core collection for the computer science area.

In contrast, the clustering paths for the items on the right side of Figure 3 are much sparser; that is, these items do not readily fall into the cluster structure in that several clustering levels are skipped, the second level corresponding to the P2 clusters being entirely absent. These items may then represent some of the principal fringe areas of the field in the sense that a substantial number of compatible items can be found in some cases, but that the concentration of items is not sufficient to define really popular subtopics.

The literature described in the cluster tree of Figure 3 is examined in the remaining sections of this study.

4. Computer Science Literature

Two principal points of view can be adopted in analyzing the clustering experiment. First the cluster structure can be considered globally in the hope of obtaining overall insight into the structure of the field. Secondly, the characteristics of individual documents may be examined to determine relationships that may exist between the clustering properties of the items and the presumed importance of these items in the field.

A) Global Cluster Structure

Consider first the analysis of the global cluster structure. Several questions may be raised: What is the scope and extent of the field? Where are the centers of gravity located and where are the fringe areas? Is some general agreement in evidence between the automatic taxonomy and the perceptions which exist among knowledgeable observers of the field? If not, what are the principal diverging areas? Does a study of the automatic classification lead to some novel viewpoints about the field.

Before answering some of these questions, it may be useful to examine the conventional attitude toward computer science. Some people still feel that a recognizable field does not exist

in the computer area, and that computing activities should be treated as applications areas for other disciplines. The majority of the observers familiar with the computer field appear however to be in agreement in recognizing at least three main subareas: [24-27

a) Theorectical foundations, including theory of computation, formal languages, complexity theory, and computer models.

b) Hardware and computer systems, including types of computers, architecture, implementation technology, system generation, and special-purpose systems.

c) Software, including programming systems, programming tools and techniques, file organization and management.

Beyond these basic components widely differing approaches exist in treating the remainder of the computer field. Consider the following topics as examples:

d) Mathematics of computing, including numerical analysis and numerical algorithms. This area is sometimes treated as a separate topic; alternatively it could be included among the basic foundations and theories of the field. Many observers prefer to consider the mathematical topics as a part of mathematics rather than computer science.

e) Special software topics such as operating systems and programming methodology. Because of their alleged scope and importance, separate subareas are often provided for such topics.

f) Data management and data base systems. This is a relatively new area considered by some people as a separate constituent of the field. Alternatively, this topic is sometimes handled as a part of the general software area, or as an application topic.

g) Methodologies valid for many different applications purposes. Greatly differing approaches may be used to treat these areas, although agreement may exist at least for some basic components such as algebraic manipulation, pattern matching, searching and sorting, and simulation and modelling.

h) Computer applications. For obvious reasons the applications areas can be treated as a part of computer science or alternatively under the various other disciplines with which they are connected. The proper classification should probably depend on the viewpoint taken in the individual documents under consideration. Normally one tends to include as a part of computer science topics for which the computational portion takes on special importance such as artificial intelligence, image processing, computer graphics, computer assisted instruction, text processing, and information retrieval.

i) Nontechnical aspects such as computer education, legal and social aspects, computer management, computer history, professional questions. Topics such as these may or may not be considered depending on one's views of the field.

Consider now the structure of the automatically derived classification based on the treatment of the sample collection under consideration. The 67 distinct P2-clusters comprising the 2,203 items of the core collection are arranged into subject areas in the display of Figure 4. 90 clusters are actually included in the Figure because some of the 67 clusters are listed several times; such duplicated clusters are identified by square boxes. The topic arrangement shown in Figure 4 is based on the classification tentatively adopted by an AFIPS committee charged with the construction of a computer science taxonomy. [24] The clusters are arranged under nine

principal headings and topic names are used to describe the individual P2-clusters in Figure 4, chosen by a conventional (nonmechani procedure after examination of the individual document titles for the items in each cluster.

The arrangement of Figure 4 as expected exhibits a substantial number of P2 core clusters under the principal areas of Theory of Computation, Computer Systems and Software. On the other hand, the strength of the Mathematics and Methodologies areas with 21 P2-clusters in each case was perhaps not obvious in advance. This tends to confirm the perceptions of those who consider these areas to be inseparable from the computer field in general.

It is perhaps not surprising that the operating systems topic under software leads in clustering strength with 12 different P2 clusters; other topics in order of clustering strength are pattern recognition (9 P2-clusters) and remote access systems (4 P2-clusters). The following additional topics that might not immediately come to mind is defining the core of the computer field all produce core clusters in the automatic classification (the number of core clusters is shown in parentheses in each case):

a) string matching algorithms (1) under Theory;

b) microprogramming (2) under Hardware;

c) compression and cryptography (1) and data structure (1) listed in the Data area;

d) community access systems (1) under Computer Systems;

e) radix conversions (1), multiprecision arithmetic (1), fast Fourier transforms (2), graph theory (2), and queueing theory (1) under the Mathematics heading (the last two topics in particular are often treated as Operations Research or pure Mathematics topics);

f) decision tables (1) under Software; and

g) theorem proving (3) and feature extraction (5) in the Methodologies area.

The left-hand side of the cluster tree of Figure 3 covers core areas that easily fit into the clustering scheme. Less central items that do not produce any standard P2-clusters are shown on the right-hand side of Figure 3. One might conjecture that the corresponding topics lie in areas that are less crucial to the field as a whole. The 2,028 bibliographic items included on the right side of Figure 3 may be separated into two classes: about 90 percent of the items (a total of 1,827) appear included in one of the small P1-clusters; the remaining 201 documents corresponding to paths, D-P3-P4, D-P4, D-P3, and D cannot readily be clustered and do not fit into either P1 or P2 clusters. A topic arrangement for the 226 P1-clusters included on the right side of Figure 3 is presented in Figure 5. Since certain clusters are again listed under several topics, the classification of Figure 5 actually includes a total of 257 P1-clusters.

The principal differences between the core areas of the literature and the fringes can be ascertained by comparing the

respective portions of Figures 4 and 5, or alternatively by examining the summary presented in Table 3. As expected, areas such as Applications and Computing Milieu are much stronger in the fringe than in the core display. All of the principal areas are represented in both listings, but the individual topic coverage is quite different. Under Mathematics of Computing, the core contains popular topics such as ordinary differential equations and fast Fourier transforms; the fringe emphasizes computer arithmetic and many subjects related to statistics and probability or operations research (transportation problem (1), knapsack problem (2), queueing theory (2), time table construction (2)). In each case there is a genuine question whether the topic should be included in the computer field.

Other differences in coverage can be identified by consulting the respective portions of the classifications. Consider the following examples:

a) Under Software, the core emphasizes the operating systems topics; in the fringe, on the other hand, the following PI-clusters are in evidence among others: programming error diagnosis (3), program verification and correctness (4), programming style and structure (3);

b) in the Data area, PI-clusters appear for coding systems (3), coding errors (4), data compression (1);

c) speech recognition (4) is present under artificial intelligence in the Methodologies area; and computer music (3) under Applications.

Had the analysis been performed with documents published in 1978 instead of 1974 or earlier, some of these topics would undoubtedly have been upgraded to become part of the core; this appears likely in particular for the previously mentioned programming error, verification, and style areas. Considering the state-of-the-art as of 1974, it is easy to agree that many of the previously listed subjects should be considered fringe rather than core areas.

Does a study of such an automatic taxonomy lead to new perceptions about the computer field? Probably not. However the classification certainly confirms that it is difficult to draw well-defined boundaries, and the strength of some of the topics normally preceived to be operations research and pure mathematics areas comes as a surprise. Overall, the conventional feelings of most people about the central areas in the computer field appear to be confirmed by the displays of Figures 4 and 5, and the existence of the several hundred fringe clusters emphasizes that the computer field is neither small nor homogeneous.

B) Individual Document Analysis

A great deal could be said about the clustering properties and the importance of individual documents. For present purposes a few typical examples must suffice. It should be recognized first of all that clustering ability on the one hand, and importance of an item in the field on the other are two distinct properties. Consider the display of Table 4: if the impact of an item in the field is assumed to be connected to some extent

with the number of citations that the item attracts, then the documents located in the bottom half of Table 4 are more important than those near the top. Unfortunately, there are many classes of important items at the bottom of Table 4 that also exhibit poor clustering properties; the reason is that good clustering characteristics are produced by homogeneous reference and citation patterns. Thus the items with the best clustering properties are those located inside the cross-hatched area of Table 4, and clearly that area is not coextensive with the good impact area consisting of the last two rows of the Table.

Among the documents that may have substantial impact in the field, but are also difficult to classify are basic reference works (many references and citations), survey and tutorial articles (many references), and interdisciplinary items (heterogeneous references and citations).

On the negative side, the correlation between citation strength and references on one hand, and impact on the other is much closer. That is, items without useful references that are unable to attract any citations are also lacking in impact. This is the case for the items in the upper left-hand corner of Table 4. Some odd documents exist in the computer area that are hard to cluster (no references or citations in the conventional sense) and for which the impact is difficult to assess. This is the case notably for programming language descriptions and various kinds of user manuals.

The interaction between clustering ability and impact may be studied further by examing a few actual sample documents in more detail. Ten items are chosen, five of which exhibit good clustering properties (labelled ① to ⑤), and five more with marginal clustering characteristics (none appear in a P2-cluster); the latter are labelled ⑥ to ⑩. The citation characteristics of these items are specified by the placement of appropriate numeric indicators in the chart of Table 4. It may be noticed that four of the five clustering items fall into the best clustering (cross-hatched) area of Table 4, while four of the five non-clustering items lie outside that area.

The full reference and citation data for the ten sample bibliographic items are included in Table 5. The bibliographic information of Table 5 is shown in an abbreviated notation adapted from the Science Citation Index, where three adjacent numbers designate volume number, beginning page number, and year of publication respectively. The citation, but not the reference, data for the documents in Table 5 are extracted from the appropriate issues of the SCI.

Most of the documents that are readily clustered contain research results with appropriate references to publications in a well-defined area. Not all of these papers are necessarily influential: when no citations are available, the homogeneous reference set can lead to the favorable clustering property. This is true notably for documents ① and ⑤ of the sample set

which carry practically no citations. Items (3) and (4) are standard research papers of some influence where both references and citations may help in the clustering process. Item (2) is a survey paper which includes a bibliography of 102 references and quite a few citations. Such a document will normally fall outside any well-defined cluster structure because of the heterogeneity of the references and citations. Item (2) provides an exception to this rule, because the survey topic is defined sufficiently narrowly in that case, and most of the references fall into a circumscribed area.

Items that cluster poorly may cover fringe areas, and quite frequently such items attract a heterogeneous clientele from many different fields. As mentioned earlier, surveys and tutorials are also difficult to cluster, and so are forgotten items that carry neither references and citations. Item (6) is an example of an influential item with quite a few citations. However, an examination of the citing journal titles listed in Table 5 shows that the citations belong to many different disciplines; this explains the clustering difficulties in this case. The next three items (numbered (7), (8) and (9)) are research papers with very few citations; the references to many unusual items, such as doctoral dissertations and internal reports, were evidently not sufficiently directed to lead these items to specific clusters.

The last item, number (10), covers research by a first-rate author published in a first-rate journal. Nevertheless, the item

belongs to the class of "lost" publications since it exhibits only one unhelpful reference and no citations in the space of three years. One can't help but feel that a little care in the assignment of references might have saved this item from oblivion.

Future analyses of the computer science literature, possibly based on a larger literature sample including a greater proportion of foreign publications, may lead to useful comparisons with the situation as of 1974 reported in present study.

REFERENCES

[1] E. Garfield, Science Citation Index -- a New Dimension in Indexing, Science, Vol. 144, No. 3619, 8 May 1964, pp. 649-654.

[2] E. Garfield, The Citation Index as a Subject Index, Current Contents, No. 18, 1 May 1974 pp. 5-7.

[3] M. M. Kessler, Comparison of Results of Bibliographic Coupling and Analytic Subject Indexing, Am. Documentation, Vol. 16, No. 3, July 1965, pp. 223-233.

[4] G. Salton, Automatic Indexing Using Bibliographic Citations, Journal of Documentation, Vol. 27, No. 2, June 1971, pp. 98-110.

[5] E. Garfield, Citation Indexes for Science, Science, Vol. 122, No. 3159, 15 July 1955, pp. 108-111.

[6] E. Garfield, Citation Indexing for Studying Science, Nature, Vol. 227, 15 August 1970 pp. 669-671.

[7] D.J. deSolla Price, Networks of Scientific Papers, Science Vol. 149, No. 3683, 1965, pp. 510-515.

[8] E. Garfield, Citation Indexing, Historio - Bibliography, and the Sociology of Science, Third Int. Congress of Medical Librarianship, Amsterdam, May 1969, pp. 187-206.

[9] E. Garfield, Citation Measures as an Objective Estimate of Creativity, Current Contents, No. 34, 26 August 1970, pp. 4-5.

[10] J.R. Cole, and S. Cole, The Ortega Hypothesis, Science, Vol. 178, No. 4059, 27 October 1972, pp. 368-375.

[11] J.R. Cole and S. Cole, Citation Analysis (Letter to Editor), Science, Vol. 183, No. 4120, 11 January 1974, pp. 32-33.

[12] W.O. Hagstrom, Inputs, Outputs and the Prestige of University Science Departments, Sociology of Education, Vol. 44, No. 4, 1971, pp. 375-397.

[13] E. Garfield, Citation Analysis as a Tool in Journal Evaluation, Science, Vol. 178, No. 4060, 3 November 1972, pp. 471-479.

[14] J. Margolis, Citation Indexing and Evaluation of Scientific Papers, Science, Vol. 155, 10 March 1967, pp. 1213-1219.

[15] E. Garfield, Is Citation Frequency a Valid Criterion for Selecting Journals, Current Contents, No. 14, 5 April 1972, pp. 5-6.

[16] J.H. Westbrook, Identifying Significant Research, Science, Vol. 132, No. 3435, 28 October 1960, pp. 1229-1234.

[17] K. Subramanyan, Core Journals in Computer Science, IEEE Trans. on Professional Communications, Vol. PC-19, No. 2, December 1976, pp. 22-25.

[18] G. Hirst and N. Talent, Computer Science Journals -- An Iterated Citation Analysis, IEEE Trans. on Professional Communication, Vol. PC-20, No. 4, December 1977, pp. 233-238.

[19] M.M. Kessler, Bibliographic Coupling Between Scientific Papers, American Documentation, Vol. 14, No. 1, January 1963, pp. 10-25.

[20] E. Garfield, ISI is Studying the Structure of Science through Cocitation Analysis, Current Contents, No. 7, 13 February 1974, pp. 5-10.

[21] H. Small and B.C. Griffith, The Structure of Scientific Literature, Identifying and Graphing Specialties, Science Studies, Vol. 4, 1974, pp. 17-40.

[22] H. Small, Cocitation in the Scientific Literature: A New Measure of the Relationship between Two Documents, Journal of the ASIS, Vol. 24, No. 4, July-August 1973, pp. 265-269.

[23] G. Salton and D. Bergmark, "Clustered File Generation and Its Application to Computer Science Taxonomies", Information Processing 77, B. Gilchrist, editor, North Holland Publishing Co., Amsterdam, 1977, pp. 441-445.

[24] AFIPS Taxonomy Committee, Taxonomy for Computer Science and Engineering, AFIPS Press, to appear.

[25] A. Ralston and C.L. Meek, Encyclopedia of Computer Science, Petrocelli/Charter Publishing Co., New York, 1976.

[26] B.W. Arden, "The Computer Science and Engineering Research Study (Cosers)", ACM Communications, Vol. 19, No. 12, December 1976, pp. 670-673.

[27] "Categories of the Computing Sciences, Classification System for Computing Review -- A Description", Computing Reviews, Vol. 17, No. 5, May 1976, pp. 175-198.

Author	Total Number of Citations from Science Citation Index* 1976	1977	Total
D.E. Knuth	186	284	470
A.V. Aho	102	144	246
J.D. Ullman	16	16	32
E.W. Dijkstra	77	121	198
C.A.R. Hoare	57	106	163
K.S. Fu	66	93	159
J.E. Hopcroft	64	90	154
H. Freeman	55	93	148
N. Wirth	35	82	117
P.J. Denning	56	42	98
J. Hartmanis	45	46	91
E.G. Coffman	54	34	88

*Self-citations are excluded

Total Number of Citations for Certain High-Frequency Authors in a Computer Science Test Collection

Table 1

4231	original documents	$\frac{4231}{4231}$ =	1.0	doc./cluster
394	pass-1 clusters	$\frac{3700}{394}$ =	9.4	docs./cluster
67	pass-2 clusters	$\frac{2203}{67}$ =	32.9	docs./cluster
30	pass-3 clusters	$\frac{2854}{30}$ =	95.1	docs./cluster
9	pass-4 clusters	$\frac{3016}{9}$ =	335.1	docs./cluster

Clustering Statistics

Table 2

	Core Areas P2 clusters	Fringe Areas P1 clusters
Theory of Computation	finite automata parsing	sequential machines formal languages concrete computational complexity
Hardware	microprogramming	
Computer Systems	networks	
Mathematics of Computation	ordinary differential equations fast Fourier transform graph theory	computer addition and multiplication matrix algebra systems of equations zeros of nonlinear equations optimization least squares statistics and probability discrete techniques graph theory
Software	structured programs storage allocation multiprogramming paging scheduling	program error diagnosis program verification, correctness program style and structure multiprocessing storage allocation
Methodologies	theorem proving feature extraction tree search	speech recognition pattern classification feature selection graphic displays
Applications	———	sciences and humanities music
Computing Milieu	———	education and computing
Data	———	coding systems coding errors

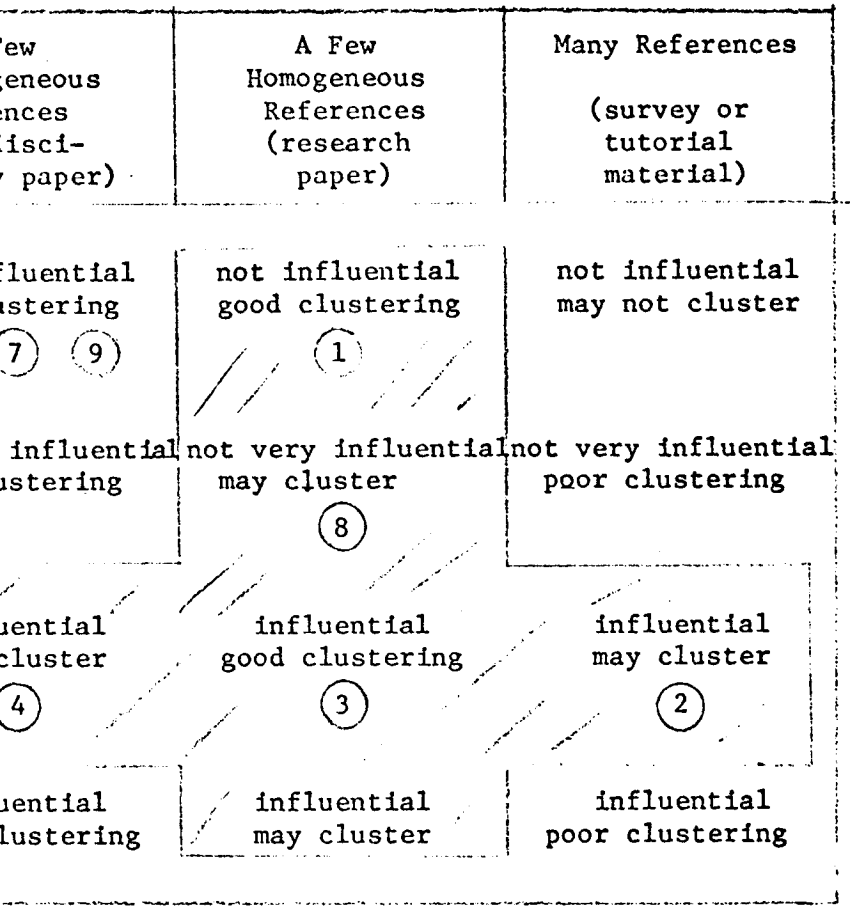
Main Cluster Concentration in Core and Fringe Areas

Table 3

References / Citations	No References (possible new area)	A F Heterog Refere (interd plinary
No Citations	lost non clustering (10)	not inf non clu (5) (
A Few Heterogenous Citations	not very influential non clustering (6)	not very non clu
A Few Homogeneous Citations	influential good clustering	influ may
Many Citations	influential may not cluster	influ poor cl

Clustering Properties and I

Tal



Impact of Bibliographic Items

ble 4

③ D. Ferrari, Improving Locality by Critical Working Sets, Communications of the ACM, Vol. 17, No. 11, November 1974, pp. 614-620.

References

L. A. Belady	IBM Systems Jr.	5	78	66
P. J. Denning	Comp. Surveys	2	153	70
A. C. McKellar	CACM	12	153	69
B. S. Brawn	CACM	13	483	70
P. Brinch Hansen	Books: Op. System Principles		190	73
L. W. Comeau	ACM Symp. on Op. Systems			67
R. F. Tsao	Book: Stat. Comp. Performance			72
D. J. Hatfield	IBM Systems Jl.	10	168	71
C. V. Ramamoorthy	Proc. ACM Nat. Comf.		229	66
T. C. Lowe	CACM	13	3	70
E. W. Ver Hoef	Proc. AFIPS '71		491	71
J. L. Baer	Proc. AFIPS '72		23	72
P. J. Denning	CACM	11	323	68
D. Ferrari	Proc. ACM Nat. Conf.		228	73
D. J. Gotthoffer	U. of Cal. Report			
H. Thompson	U. of Cal. Report			

Citations

1975	P. J. Denning	Proc. IEEE	63	924	75
	D. Ferrari	IBM Jl. Research	19	244	75
1976	D. Ferrari	Computer	9	36	76
	A. W. Madison	CACM	19	285	76
1977	A. Sangiora	IEEE Trans. on Circuits	24	709	77

④ J. Bruno, R. Sethi, E. G. Coffman, Scheduling Independent Tasks to Reduce Mean Finishing Time, Communications of the ACM, Vol. 17, No. 7, July 1974, pp. 382-387.

References

R. W. Conway	Book: Theory of Scheduling			67
L. R. Ford	Book: Flows in Networks			62
J. A. Bruno	Penn. State U. Report			73
S. A. Cook	3rd ACM Symp. on Theory Comp.		151	71
R. M. Karp	Book: Complexity of Comp.		85	72
S. Sahni	13th Symp. on Switching		130	72
J. D. Ullman	U. of Cal. Report			73
R. L. Graham	Bell Syst. Tech. Journal		1563	66
R. L. Graham	SIAM Jl. of Applied Math.	17	416	69
E. G. Coffman	Penn. State U. Report			73

Citations

1975	K. L. Krause	JACM	22	522	75
	J. D. Ullman	Jl. Comp. Syst. Science	10	384	75
1976	J. Bruno	Jl. Comp. Syst. Science	12	319	76
	E. G. Coffman	Acta Informatica	6	1	76
	M. J. Gonzales	IEEE Trans. Aerospace	12	530	76
	E. Horowitz	JACM	23	317	76
	S. K. Sahni	JACM	23	556	76
	S. K. Sahni	JACM	23	116	76
1977	M. R. Garey	Book: Approximate Alg.		41	76
	D. H. Ibarra	JACM	24	280	77

Table 5 (cont.)

⑤ E. Horowitz, A Sorting Algorithm for Polynomial Multiplication, Journal of the ACM, Vol. 22, No. 4, October 1975, pp. 450-562.

References

A. V. Aho	Book: Design Analysis Alg.			74
M. Fredman	7th ACM Symp. Theory of Comp.		240	75
E. Horowitz	Proc. Math. Software		45	74
S. C. Johnson	Proc. Euro Sam SIGSAM Bull.	8	63	74
D. E. Knuth	Book: Art of Comp. Prog. Vol. 2			71
D. E. Knuth	Book: Art of Comp. Prog. Vol. 3			71

Citations

1976 -

1977 -

1978 -

⑥ A. K. Cline, Scalar and Planar Valued Curve Fitting Using Splines under Tension, Communications of the ACM, Vol. 17, No. 4, April 1974, pp. 218-220.

References

A. K. Cline		CACM	17	218	74

Citations

1975	-				
1976	G. M. Nielson	SIAM Review	18	820	76
	S. Pruess	Jl. of Approx. Th.	17	86	76
	D. H. Thomas	Math. of Comp.	30	58	76
1977	W. E. Baker	Monthly Weather Rev.	105	1384	77
	I. C. Csizmadi	Jl. of Chem. Soc. Trans. 2		542	77
	H. P. Jones	Jl. of Quant. Spectroscopy	17	765	77
	R. C. Phillips	Book: Software Tools		205	76
	K. C. Yeh	Jl. of Pharm. Sciences	66	1688	77

Table 5 (cont.)

⑦ P. S. Wang, The Undecidability of the Existence of Zeros of Real Elementary Functions, Journal of the ACM, Vol. 21, No. 4, October 1974, pp. 586-585.

References

B. F. Caviness	JACM	17	385	70
M. Davis	Am. Math. Monthly	80	233	73
D. Hilbert	Bull. Am. Math. Soc.	8	437	01
Iu. V. Matiasevitch	Doklady Akad. Nauk.	191	279	70
J. Moses	CACM	14	548	71
D. Richardson	Jl. Symbolic Logic	33	514	68
R. Risch	Bull. Am. Math. Soc.	76	605	70
P. S. Wang	MIT Doctoral Dissertation			71
--	Macsyma Reference Manual			74

Citations

1975	-				
1976	-				
1977	A. D. McGettri	Computer Jl.	20	263	77

⑧ I. H. Sudborough, A. Note on Tape Bounded Complexity Classes and Linear Context Free Languages, Journal of the ACM, Vol. 22, No. 4, October 1975, pp. 499-500.

References

S. Ginsburg	SIAM Jl. on Control	4	429	66
S. A. Greibach	SIAM Jl. on Computing	2	304	73
N. D. Jones	U. of Kansas Report			73
T. Kasami	Info. and Control	10	209	68
P. M. Lewis	IEEE Conf. on Switch. Circuits		191	65
A. R. Meyer	ACM Symp. on Theory of Comp.		1	73
W. J. Savitch	Jl. Comp. System Science	4	177	70
I. H. Sudborough	Jl. Comp. System Science	10	62	75

Citations

1976	-				
1977	N. D. Jones	Info. and Control	35	177	77
	B. Monien	Acta Informatica	8	377	77
1978	L. Boasson	Math. Systems Theory	11	147	77
	I. H. Sudborough	Int. Jl. of Comp. Math.	6	191	77

Table 5 (cont.)

(9) C. J. Van Rijsbergen, Best Match Problem in Document Retrieval, Communications of the ACM, Vol. 17, No. 11, November 1974, pp. 648-649.

References

W. A. Burkhard	CACM	16	230	73
D. Crouch	SMU Doctoral Thesis			73
L. Hyrarinan	Nord. Tidskr. Inf. Behand.	2	83	62
N. Jardine	Info. Storage Retrieval	7	217	71
B. Litofsky	U. of Penn Doctoral Thesis			69
J. Minker	Info. Storage Retrieval	8	329	72
D. M. Murray	Cornell U. Doctoral Thesis			72
G. Salton	Book: Auto. Info. Sto. Retr.			68
C. J. Van Rijsbergen	Computer Journal	14	407	71
C. J. Van Rijsbergen	U. of Cambridge Doctroal Thesis			72

Citations

1975 -

1976 -

1977 -

(10) E. W. Dijkstra, Self Stabilizing Systems in Spite of Distributed Control, Communications of the ACM, Vol. 17, No. 11, November 1974, pp. 643-644.

References

C. S. Scholten — Private Communication

Citations

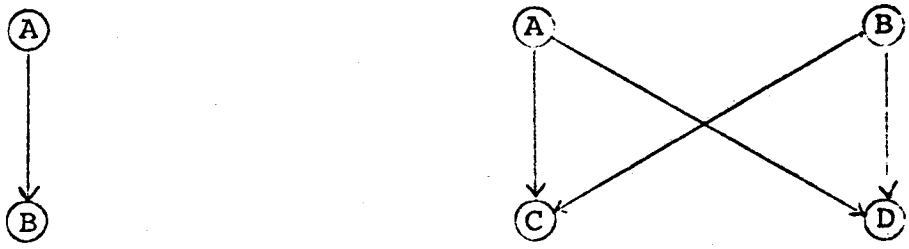
1975 -

1976 -

1977 -

Citation and Reference Data for Sample Documents

Table 5

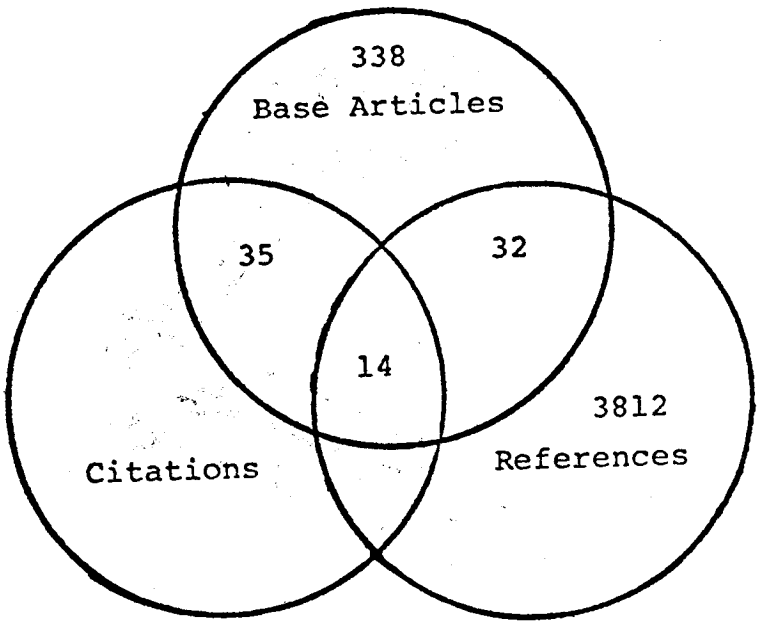


a) Item A refers to B; item B is cited by A.

b) Items A and B are related by bibliographic coupling; items C and D are related by cocitations

Bibliographic Coupling and Cocitation Patterns

Figure 1



419 Base Articles

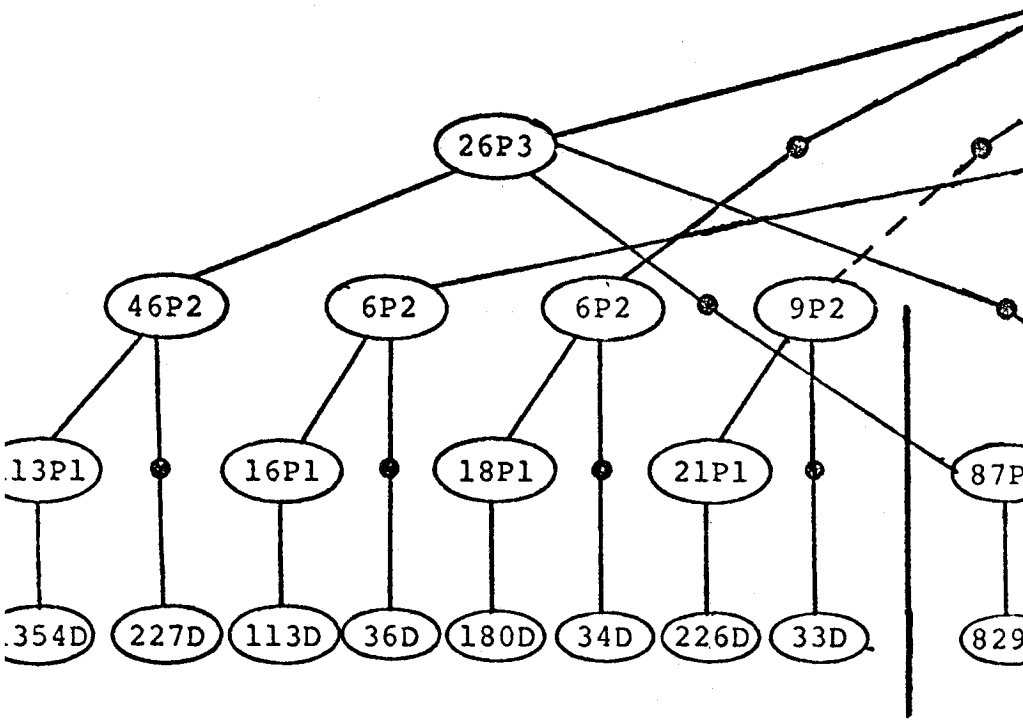
3858 Distinct References from Base Articles

4231 Distinct Collection Items

4764 Distinct Links between Items

Experimental Computer Science Collection

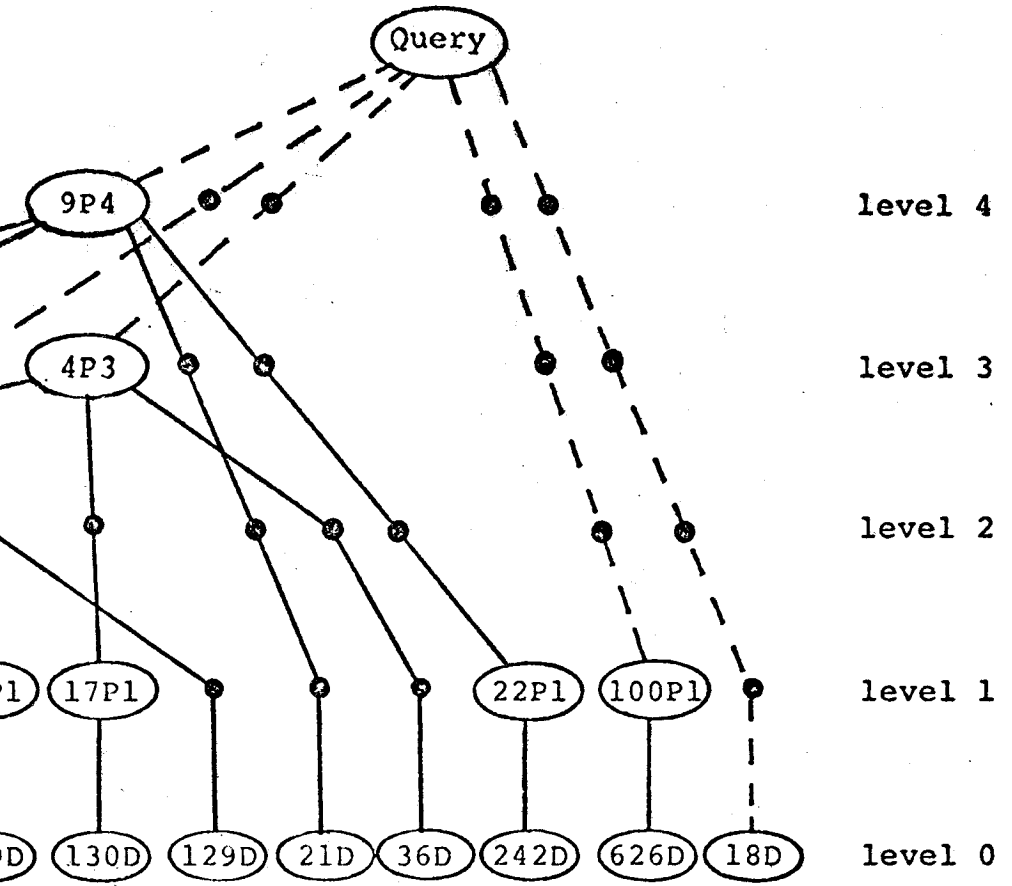
Figure 2



Easy Clustering 52%
2203 items

Computer Scienc

Fig.



Hard Clustering 48%
2028 items
(1827 P1 items 43%; 201 D items 5%)

ce Cluster Tree

. 3

Theory of Computation

switching and automata
  finite automata
  switching functions

formal languages
  syntactic complexity
  parsing

logic of computer programs
  program optimization

formal program semantics

complexity of programs

computer models
  machine theory
  Turing machines

abstract computational complexity

concrete computational complexity
  string matching

Hardware

types of computers
  multiprocessors

computer subsystems

peripheral devices

data entry and terminals

hardware design
  microprogramming

performance and reliability

implementation technology

computer architecture

Computing Milieux

economics of computing

education and computing

history and computing

legal aspects

management of computing

computing profession

social effects of computing
  privacy log-in
  community systems

Data

structure

representation

communication

compression
  cryptography

Computer Systems

structure based systems
  multiprocessors
  networks

remote access systems
  time sharing
  log-in process
  community systems
  decision support

special purpose systems

in-out and memory systems

applications-based systems
  (air traffic, point of sale)

Mathematics of Computing

computer arithmetic
  floating point
  multi-precision
  radix conversion

linear equations and algebra
  inversion
  linear systems

zeros of nonlinear equations

numerical integration and differentiation

ordinary differential equations

partial differential equations

interpolation, approximation
  fast Fourier transform
  spline functions

optimization
  nonlinear systems

mathematical programming
  integer programming
  linear programming

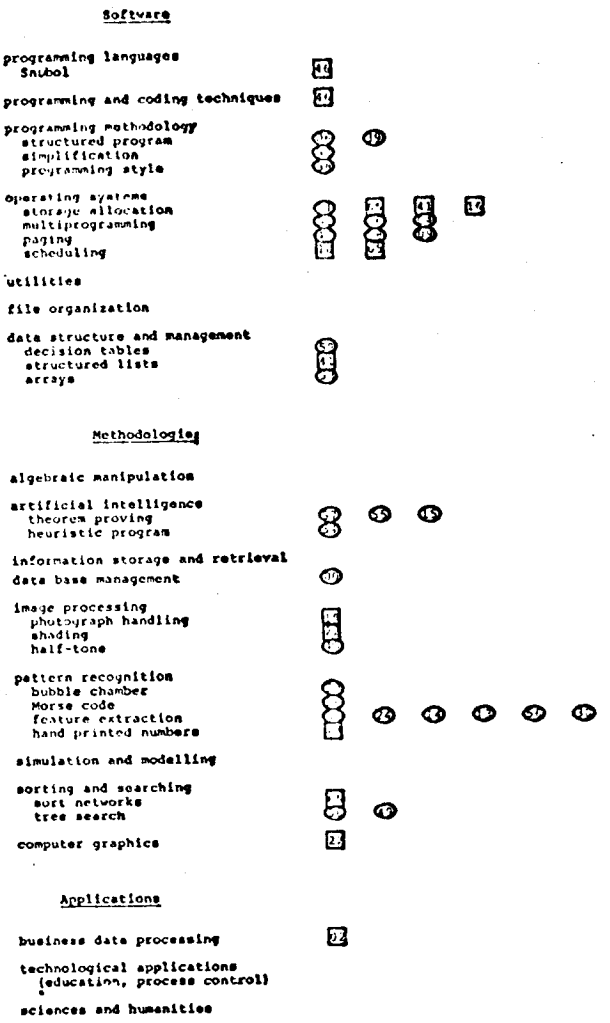
algorithm selection

numerical software

statistics and probability
  factoring
  random variable

discrete mathematics
  graph theory
  queueing theory

Computer Science Taxonomy--Easy-To-
Pass-2 clusters
(2203 items, 67 distinct clusters, unique



Cluster Topics

(○, duplicated □)

Computer Science Taxonomy - Har
Pass-1 cluste
(1827 distinct items, 226 distinct clusters,

Fig. 5

rd-to-Cluster Topics
ers
, unique ⬭, duplicated ▭)

Theory of Computation

- switching and automata
  - switching theory [1355]
  - switching circuits (3776)
  - propositional calculus (3347)
  - asynchronous logic (4851)
  - logic primitives (3754) (3755)
  - threshold logic (3843)
  - Boolean equations (3741)
  - sequential machines (3844) (3452) (3414) [3839]
  - finite state automata [3742] [3539]
  - stochastic automata (3800) (3757)
  - pushdown automata (3394)
  - fuzzy automata (5851)
  - decomposition (3830)
  - equivalence in automata (3827)
  - machine complexity (3743)
- formal languages (1010) (1198) (2571)
  - context free languages (4755) (5948)
  - ambiguity [3742]
  - finite languages [3539]
  - precedence grammar (2453) (3756)
  - ATN grammar [1664]
- logic of computer programs [3333]
  - correctness [3839]
  - program optimization (5505)
- formal semantics of computer programs [1007]
- complexity of programs [2572]
  - program minimization (1009)
- computer models
  - Turing machines [4755]
- abstract computational complexity [1007] (1348)
- concrete computational complexity (4711) [2194] [1642]

Applications

- business data processing
- technological applications (CAI) (2508)
- sciences, humanities (5326) (3798) (3795)
  - music (5670) (5261) (1665)
  - mathematical linguistics (5673)
  - EKG patterns [3822]

Hardware

- types of computers
- computer subsystems
  - memory (3834)
- peripheral devices
- data entry and terminals
- hardware design (5774)
  - microprogram (5771)
  - microprocessor [3797]
- performance and reliability (5252)
  - memory errors [3838]
  - fault tolerant computing (3797)
  - fault location (5773)
- implementation technology
  - LSI technology (5769)
- computer architecture [2258]

Computer Systems

- structure based systems
  - networks (5672)
  - parallel systems (3753)
  - array storage (3836)
- remote access systems
  - time sharing [2258] [5938]
  - distributed systems (1667)
  - message switching (5116)
- special purpose systems
  - raster display [2581]
- in-out and memory systems
- applications-based systems
  - signal processing [3737]

Computing Milieu

- economics of computing
- education and computing (2507) (2505) (3643)
- history of computing
- legal aspects
- management of computing
- computing profession (5330)
- social effects

Mathematics of Computing

- uter arithmetic — 3769 3767
- dix systems — 5451
- oating point — 3777 3801 3845 3745
- d, multiply, divide — 3846 3848
- uare roots — 5449
- undoff error

- ar equations and algebra — 3748 1347
- trix algebra — 5378 3307 4788 3765
- stems of equations — 3484 2193 4858
- groups — 3764
- ries computation — 5379 5324
- tensors — 3581
- completeness — 3474

- s of nonlinear equations — 3338 2516 2509
- rical integration and differentiation — 3525 2195
- inary differential equations — 2450
- tial differential equations

- erpolation, approximation — 2366
- ast Fourier transform — 3734 3764
- plines — 5377
- urve fitting — 2513

- imization — 3501 2364 5846
- east squares — 5263 3547 3492

- hematical programming — 3501
- ransportation problem — 4867
- napsack problem — 5254 4866

- orithm selection

- erical software

- tistics and probability — 5669 2200 5663
- athematical statistics — 2931
- random numbers — 2932 3737
- stochastic process — 3547

- screte mathematical techniques — 2361 2362 3340
- graph theory — 1345 2205 2357 4855
  4857 3751 1344 3490
  1346 3497 3729

- queueing theory — 5503 5143
- time table construction — 5844 2929
- game theory — 3603
- combinatorics — 2515

Software

- programming languages
  - APL — 1498
  - Algol — 2511 5940
  - Simula — 3329
  - compilers — 3588 5446
  - data types — 5447
  - control structures — 5940
- programming and coding techniques
  - recursion — 2511
  - error diagnosis — 3347 5119 2457
- programming methodology
  - verification, correctness — 5938 5008 3333 2572
  - style, structure — 5939 2924 2639
  - design — 1671
  - optimization — 5504
  - parallel execution — 2570
  - operation code interpretation — 3732 3767
- operating systems
  - multiprocessing — 2358 2926 3785 4689
  - storage allocation — 5254 5452 3765 5253
    1666 4862
  - scheduling — 2927 9115
  - spooling — 5603
  - segmenting — 5499
  - Multics system — 2567
  - time sharing system — 5941
- utilities
  - statistical packages — 2210 3388
- file organization — 2567
- data structures and management
  - decision tables — 2497 2031

Data

- data structure — 3733 3577
- data representation
  - coding systems — 3509 3485 5118
  - coding errors — 3410 4764 3838 3797
- data communication
- data compression — 2353

Methodologies

lgebraic manipulation (2829) (4690)

rtificial intelligence (3803)
- speech recognition (3812) [3818] [3804] [1664]
- theorem proving [1358] (5847)
- robotics (5667)
- question answering [2936] (5848)
- A.I. languages (1558)

ıformation storage and retrieval [1670]
- file organization [3828]
- text searching (2204) (3610)
- relevance (3616)
- automatic classification [2936]
- automatic indexing (4860)

ıta base management [5118]

ıage processing (3760)
- compression (3752)
- enhancing (3823)
- hidden line (5256)

ıttern recognition
- classification [2194] (3730) (3763) (3840)
- classification error (3832)
- feature selection (3770) (3809) (3810) (3813) (3815) (3819)
- speech features [3804] [3818]
- stochastic models (3791)
- syntactic models (5664)
- EKG patterns [3822]
- chromosome patterns (3807)
- scaling (3842)

ımulation and modelling (5497) (5607)

ɔrting and searching
- best match (5852) [1670]
- binary search [3828] [1642]
- text search (2204)

ɔmputer graphics
- displays [2581] (2452) (3634)
- polyhedra (5666)