The Use of Extended Boolean Logic in Information Retrieval*

Gerard Salton

TR 84-588
January 1984

Department of Computer Science
Cornell University
Ithaca, New York 14853

*This study was supported in part by the National Science Foundation under grant IST 81-08696.

The Use of Extended Boolean Logic in Information Retrieval

Gerard Salton*

Abstract

An extended Boolean retrieval strategy has previously been introduced in which the individual Boolean operators can be treated more or less strictly, depending on the perceived strength of association of the query terms. The extended Boolean system is illustrated by examples and evaluation output is used to demonstrate the effectiveness of the operations.

*Department of Computer Science, Cornell University, Ithaca, NY 14853. This study was supported in part by the National Science Foundation under grant IST 81-08696.

1. Conventional Boolean Information Processing

Boolean logic is used widely in information processing to formulate queries submitted to data base or bibliographic retrieval services, and to specify alternative paths or procedures in artificial intelligence programs. In the retrieval context, the search requests consist of sets of attribute-values, or sets of content terms, interrelated by the Boolean operators and, or, not. The records to be retrieved are often identified by using an "inverted" term (or attribute-value) index which provides for each query term a list of the record identifiers carrying the corresponding term. To determine the set of items responding to a query such as (A and B), it is then only necessary to intersect the A and B lists of record identifiers from the inverted index; correspondingly, to identify the items responding to query (A or B), a list union operation is performed.

The conventional Boolean retrieval methodology is not well adapted to the information retrieval task for the following reasons:

a) The formulation of good Boolean queries is an art rather than a science; most untrained users are unable to generate effective query statements without assistance from trained searchers.

b) The standard Boolean retrieval methodology does not provide any direct control over the size of the output; some query statements may provide no output at all, whereas other statements provide an unmanageably large number of retrieved items.

c) The Boolean methodology does not provide a ranking of the retrieved items in any order of presumed usefulness, thus all retrieved items are presumed to be equally good, or equally poor, for the user.

d) The Boolean system does not provide for the assignment of weights to the terms attached to documents or queries; thus each assigned term is assumed to be as important as each other assigned term, the only distinction actually made is between terms that are assigned (with an implied weight equal to 1), and terms that are not assigned (with an implied weight equal to 0).

e) The standard retrieval methodology may produce results which appear to be counter intuitive:

i) in response to an or-query (A or B or ... or Z) a record or document with only one query term is assumed to be as important as a document containing all query terms;

ii) in response to an and-query (A and B and ... and Z) a document containing all but one of the query terms is considered as useless as a document with no query term at all.

In fact if term weights of 1 and 0 are used to distinguish terms that are respectively present in, or absent from, a given document specification, then retrieval of a given document depends only on the term of maximum weight for or-queries and the term of minimum weight for and-queries. When the term of maximum weight is equal to 1 for or, then at least one term is equal to 1, and the item is retrieved; correspondingly when the term of minimum weight is equal to 1 for and, then all terms are equal to 1, and the item is retrieved.

The vector processing system of retrieval does not exhibit most of the disadvantages of the Boolean query system. In the vector model a search

request consists of a simple, possibly weighted, set of terms $Q = (q_1, q_2, \ldots, q_n)$ without any Boolean operators. Since the records and documents are also characterized by attribute-values or terms -- for example, $D_i = (d_{i1}, d_{i2}, \ldots, d_{in})$, it becomes possible to compute a similarity measure between the query and each document, such as

$$sim(D_i, Q) = \sum_{j=1}^{n} q_j \cdot d_{ij} \quad . \tag{1}$$

Following the calculation of the query-document similarities, the documents can be ranked in decreasing order of the similarity coefficients. This provides appropriate output control since a user can retrieve only a few top ranked items. In the vector processing system, the query formulations are relatively easy to generate since it becomes unnecessary to choose Boolean operators to relate the terms; furthermore, variable term weights can be accommodated. Unfortunately, the query structure inherent in the Boolean system is no longer present. This implies that no distinction is made between the use of synonyms related by or, such as "microcomputers or minicomputers or hand-held calculators", and the use of term phrases related by and such as "information and retrieval".

In the remainder of this note a generalized extended Boolean retrieval system is introduced which is compatible both with the conventional Boolean system and with the vector processing system. The extended system thus provides ranked output control and term weighting while still preserving the query structure inherent in the Boolean system. The extended system also produces much better retrieval output than the conventional methodologies.

2. Extended Boolean Information Retrieval

If one assumes, as before, that the presence of a term in a document represents a value of 1 and the absence of a term from a document a value of 0, then a given term assignment to a document may be represented by a binary vector of the form (1,0,0,1,0...). For the previously given sample vector, the 1's in positions 1 and 4 indicate that terms 1 and 4 are assigned to the document, whereas the 0's in positions 2, 3, and 5 show that the corresponding terms are absent from the document. Assuming that n terms are usable for document identification, a given document is then representable by an n-dimensional vector.

Consider now an or-query: the only document not retrieved by such a query in a Boolean system is the document consisting of all 0's, that is, D = (0,0,...,0). This suggests that the goodness of a document with respect to an or-query may be computed as a function of the distance from the 0 point in the vector space. Analogously, with respect to an and-query, the only retrieved document is the one containing all the terms, that is, D = (1,1,1,...,1). Thus the goodness of an item for an and-query is computed as a function of the similarity to the 1 point in the document space.

The following distance, or similarity, measures may be used for this purpose:

a) Consider the two-term queries (A or B) and (A and B). Assuming that the query terms are not weighted but that document term weights d_A and d_B ranging from 0 to 1 reflect the importance of terms A and B in the document, the following measures may be used to reflect the retrieval value of a document with respect to the two Boolean queries:

$$\text{sim}(D,Q_{A \underline{\text{or}} B}) = \left[\frac{d_A^p + d_B^p}{2}\right]^{1/p} \qquad (2)$$

$$\text{sim}(D,Q_{A \underline{\text{and}} B}) = 1 - \left[\frac{(1-d_A)^p + (1-d_B)^p}{2}\right]^{1/p}$$

b) If one assumes that the query terms are also weighted using weights a,b for query terms A and B respectively, the similarity expressions (2) now become

$$\text{sim}(D,Q_{[(A,a) \underline{\text{or}} (B,b)]}) = \left[\frac{a^p d_A^p + b^p d_B^p}{a^p + b^p}\right]^{1/p} \qquad (3)$$

$$\text{sim}(D,Q_{[(A,a) \underline{\text{and}} (B,b)]}) = 1 - \left[\frac{a^p(1-d_A)^p + b^p(1-d_B)^p}{a^p + b^p}\right]^{1/p}$$

When the values of a and b are reduced to 0 and 1, the expressions (3) reduce to the simpler similarity functions (2) used earlier for the unweighted case.

c) When the documents and queries are identified by n terms, instead of 2 terms, the expressions (3) are simply lengthened by using weights c,d,e,... and d_C, d_D, d_E... for terms C,D,E ... assigned to queries and documents, respectively.

d) For mixed queries containing both <u>and</u> and <u>or</u> operators, the two expressions in (2) and (3) are appropriately combined. For example for query Q = {(A,a) <u>or</u> (b[(E,e) <u>and</u> (F,f)])} the appropriate retrieval value for a document D with term values (d_A, d_E, d_F) will be

$$\text{Sim}(D,Q) = \left\{ \frac{a^p d_A^p + b^p \left[1 - \frac{(e^p(1-d_E)^p + f^p(1-d_F)^p)^{1/p}}{e^p + f^p}\right]^p}{a^p + b^p} \right\}^{1/p} \qquad (4)$$

In the expressions (2), (3), and (4) the value of the parameter p determines the interpretation of the Boolean operators. The following results are easy to prove [1,2]:

a) When p=1, the results for or and and are identical ($sim(D,Q_{or}) = sim(D,Q_{and})$); that is, no distinction is made between or and and, and the result is similar to that obtained in a vector processing system by the inner product similarity of expression (1).

b) When p=∞, the results obtained are compatible with those produced by a conventional Boolean retrieval system. Thus $sim(D,Q_{or}) = max(d_A,d_B,d_C,...)$ and $sim(D,Q_{and}) = min(d_A,d_B,d_C,...)$ when p=∞ and the query terms are unweighted. This is the exact result obtained with normal Boolean retrieval.

c) When p is reduced from ∞ to 1, the results produced are intermediate between a pure Boolean system and a vector processing system. That is, a strict phrase (A and B) is interpreted as a fuzzy, tentative phrase; and a set of strict synonyms (A or B or C) is interpreted as a set of approximately similar terms.

Since the pure Boolean system may be too strict in practice while the vector processing system is too lax, intermediate p-values may give optimum results. In that case, the lower the p-value assigned to an operator, the less strict will be the interpretation of the corresponding Boolean operator.

The effectiveness of the extended Boolean operations is illustrated by

using two collections of documents represented by bibliographic citations and abstracts. The CACM collection consists of 3204 articles in computer science, published in ACM Communications between 1958 to 1978, and the CISI collection covers 1460 articles in automatic documentation. These collections are used with 52 and 35 search requests, respectively. The results of Table 1 give the search precision values (the proportion of retrieved materials actually found relevant) averaged over the corresponding query set, and computed at certain fixed values of the search recall (the proportion of relevant materials actually retrieved). At the bottom of each portion of Table 1, an overall precision result is given for each run, consisting of the average of three precision values computed at low, medium and high recall levels of 0.25, 0.50, and 0.75, respectively.

Four different retrieval runs are illustrated in Table 1, including the conventional Boolean methodology ($p=\infty$), and three runs in the extended system using uniform p-values of 1 and 2, respectively, and mixed p-value assignments where more important term combinations are assigned p-values close to 2, and less important term combinations are identified by p-values close to 1. For the conventional Boolean runs, the query and document terms were not weighted, or equivalently each term received a weight of 1. In the extended Boolean system, automatically computed term weights in inverse document frequency (idf) order were used for the query terms. In the idf term weighting system the highest weights are obtained for terms occurring least frequently in the documents of a collection. [3] For the document terms, the assigned weight is a combination of the term frequency (the number of times a term occurs in the document) and the inverse document frequency. [3] The terms used for document identification were extracted automatically from the texts of the document titles and

abstracts, and the Boolean queries were manually formulated from originally available natural language statements of user need.

The results of Table 1 show that the conventional Boolean methodology produces by far the worst performance. The improvements for the extended system reach 76 percent for CACM and 58 percent for CISI. In each case, the pure weighted vector system (p=1) produces a high standard of performance, which is then further improved by using weak term combinations represented by p=2 and mixed p-values. To be able to compute the recall-precision values of Table 1, where continuous variations of precision are given corresponding to specific changes in recall, ranked retrieval output should be available so that the documents may be considered in decreasing order of presumed relevance to the queries. In the extended system, the search output is automatically ranked in decreasing order of the query-document similarity. In the conventional Boolean system, only two sets of items are normally distinguished--those retrieved with an implied query-document similarity of 1, and those not retrieved with a query-document similarity of 0. For the current experiments, the n items retrieved by the conventional Boolean system were assigned random ranks from 1 to n, whereas the remaining N-n nonretrieved items received random ranks from n+1 to N.

3. Discussion

To understand the the detailed operations of the extended Boolean system it is necessary to look at individual query operations. Consider the four queries CISI 1, CACM 5, CACM 21 and CACM 40 illustrated in Table 2. The original natural language formulation is shown in Table 2 in each case, together with the conventional, manually constructed Boolean formulation.

The p-values used in the mixed-p case are shown as superscripts attached to the Boolean operators in the query formulations of Table 2. For the conventional Boolean runs these values are replaced by uniform settings of $p=\infty$; additional uniform settings of p=1 and p=2 are used for the two other experiments.

Six sample documents are reproduced in Appendix A, numbered I 2932, I 2703, I 35, I 2941, I 2956 and I 1502. Each of these documents was judged to be relevant to one of the sample queries of Table 2. For each document, the appendix shows the full text used for term assignment by the automatic indexing system. For some of the CACM documents this includes titles (.T), abstract (.W), CACM issue number (.B), author names (.A), manually assigned keywords (.K), and Computing Reviews categories (.C).

The advantages of the extended system over the conventional Boolean method stems from two main causes:

a) a better discrimination of the retrieved items in the extended system where the computed retrieval ranks are based on <u>all</u> the document terms which match a portion of the query rather than or only one term, or one clause, as in the conventional Boolean system;

b) a much greater facility of retrieving early in a search relevant items which exhibit substantial similarity with the Boolean query, but which are nevertheless rejected by the conventional Boolean logic.

Consider first documents I 2932 and I 2703. Both items are judged relevant to query CACM 21. The table of retrieval ranks and query-document

similarities for these two documents (Table 3(a)) shows that these two items are actually retrieved by the conventional Boolean system (query-document similarity equal to 1) because of the combined presence of "complex" and "computation" in the document texts. However neither document would have been likely to be noticed by the user in a conventional system because of the assigned retrieval ranks of 130 and 82. (The retrieval output obtained in response to query 21 contains some relevant items but also a large number of items that match the query but are not relevant to the user need; many of these items are retrieved ahead of items 2703 and 2932.)

When p-values smaller than infinity are used, all the matching query terms count, including for example "tractability" in I 2932 and "algorithm" in I 2703. Furthermore, the weights of the matching terms also play a role in determining the query-document similarities, and hence the retrieval ranks. Both "computation" and "complex" and "algorithm" all occur twice in I 2703. This accounts for the fact that very low retrieval ranks are obtained for both of these items when low p-values are used. For I 2932, the best output rank of 4 corresponds to p=2, where the matching term combination of "computation" and "complex" is given extra weight. Document 2703 is actually the first retrieved document for very low p (p-mixed and p=1).

Document I 35 is judged relevant to query CISI 1. Unfortunately the required query term "title" is not present in the document text. Hence this document is rejected in the conventional Boolean system as shown in Table 3(b). Since several other query terms are actually present in the document ("content", "relevance"), much improved retrieval ranks are obtained when p=2 where the combination of "content or relevance" is given

extra weight.

The same phenomenon is noticeable for document I 2941 and I 2956 which are relevant to query CACM 40. In both cases the documents are rejected by the conventional Boolean run, the output ranks being 1000 and 1720, respectively. Since the query word "type" occurs several times in the document texts, these documents receive much improved retrieval ranks for lower p-values. For I 2941 and I 2956, the best p value is 1 (corresponding to retrieval ranks of 15 and 18, respectively), because the word "type" alone determines the retrieval rank, rather than a combination of several matching query terms.

In a reformulated search request where the phrases "data type" (data and type) and "programming language" (program and language) would occur explicitly, document 2941 can actually be retrieved by a conventional Boolean search, and additional improvements are obtainable in the extended system. Table 3(d) shows the retrieval ranks for a reformulated query 40:

"((data and type) and (pascal or clu or alphard
or russell or ada or (algol and 68) or ell
or (program and language))."

When p=1, the two items are retrieved by the new query formulation within the top ten items. But in the conventional Boolean system the retrieved item 2941 receives a retrieval rank of 15; item 2956 is not retrievable even by the reformulated query using the conventional methodology.

Consider, as a final example, document I 1502 which is judged to be relevant to query CACM 5. Table 4 shows the retrieval ranks of the 8 documents which are relevant to that query for the conventional Boolean system

($p=\infty$), and for the extended system when $p=2$. It may be seen that only 1 relevant out of 8 is retrieved using the conventional system (item 2035). The sample document 1502 lacks the necessary terms "satisfaction" or "efficiency" and is therefore rejected. The document does, however, exhibit multiple occurrences of the query term "edit", coupled with an occurrence of the query term "user". This accounts for the much improved retrieval rank of 3 when $p=2$. Indeed, 5 of the 8 relevant documents are retrieved in the extended system within the top 10 ranks when $p=2$.

The foregoing illustrations demonstrate that the relaxed Boolean logic represents a powerful tool in bibliographic retrieval systems. The extended logic should also be adaptable to data base processing and to certain artificial intelligence applications.

References

[1] G. Salton, E.A. Fox and H. Wu, Extended Boolean Information Retrieval, Communications of the ACM, Vol. 26, No. 11, November 1983, p. 1022-1036.

[2] G. Salton, C. Buckley and E.A. Fox, Automatic Query Formulations in Information Retrieval, Journal of the ASIS, Vol. 34, No. 4, July 1983, p. 262-280.

[3] G. Salton and M.J. McGill, Introduction to Modern Information Retrieval, McGraw Hill Book Company, New York, 1983.

Search Precision at Specified Recall Level
Averaged Over Query Set

Recall	Conventional Boolean p=(∞)	All p-values uniform (p=1)	All p-values uniform (p=2)	Variable p-values (1≤p≤2)
0.1	.3709	.5939	.6071	.5944
0.2	.3066	.5195	.5088	.5116
0.3	.2360	.3991	.4093	.4171
0.4	.2238	.3577	.3675	.3748
0.5	.2021	.2818	.3121	.3159
0.6	.1624	.2472	.2690	.2723
0.7	.0909	.1730	.1839	.1868
0.8	.0812	.1319	.1444	.1453
0.9	.0632	.1071	.1164	.1166
1.0	.0632	.0771	.0870	.0870
overall	.1798	.3016	.3131	.3166
		+67.2%	+74.1%	+76.1%

a) CACM Collection (3204 documents, 52 queries)

0.1	.2313	.3931	.3766	.3787
0.2	.1980	.3133	.3073	.3179
0.3	.1426	.2202	.2257	.2280
0.4	.1225	.1657	.1720	.1722
0.5	.1112	.1491	.1547	.1563
0.6	.0828	.1306	.1304	.1317
0.7	.0610	.1119	.1136	.1156
0.8	.0505	.0902	.0847	.0870
0.9	.0416	.0697	.0653	.0678
1.0	.0357	.0435	.0433	.0433
overall	.1118	.1726	.1728	.1771
		+54.4%	+54.6%	+58.5%

b) CISI Collection (1460 documents, 35 queries)

Search Effectiveness of Extended Boolean Retrieval System

Table 1

CISI 1 (46 relevant items)

What problems and concerns are there in making up descriptive titles?
What difficulties are involved in automatically retrieving articles from approximate titles?
What is the usual relevance of the content of articles to their titles?

(title $\underline{and}^{2}$ (automatic $\underline{or}^{1.5}$ retrieve $\underline{or}^{1.5}$ problem $\underline{or}^{1.5}$ concern $\underline{or}^{1.5}$ descriptive $\underline{or}^{1.5}$ approximate $\underline{or}^{1.5}$ difficulty $\underline{or}^{1.5}$ content $\underline{or}^{1.5}$ relevance $\underline{or}^{1.5}$ article))

CACM 5 (8 relevant items)

I'd like papers on design and implementation of editing interfaces, window-managers, command interpreters, etc. The essential issues are human interface design, with views on improvements to user efficiency, effectiveness and satisfaction.

(editing $\underline{and}^{1.2}$ (human $\underline{or}^{1.5}$ user) $\underline{and}^{2}$ (satisfaction $\underline{or}^{1.5}$ efficiency))

CACM 21 (11 relevant items)

Computational complexity, intractability, class-complete reductions, algorithms and efficiency.

((computational $\underline{and}^{2}$ complexity) $\underline{or}^{1.5}$ tractability $\underline{or}^{1.5}$ (class $\underline{and}^{1.5}$ complete $\underline{and}^{1.5}$ reduction) $\underline{or}^{1.5}$ (algorithm $\underline{and}^{1.7}$ efficiency))

CACM 40 (10 relevant items)

List all articles dealing with data types in the following languages: Pascal, CLU, Alphard, Russell, Ada, ALGOL 68, Ell. List any other languages that are referenced frequently in papers on the above languages (e.g. catch all languages with interesting type structures that I might have missed).

(type $\underline{and}^{1.7}$ (pascal $\underline{or}^{1.5}$ clu $\underline{or}^{1.5}$ alphard $\underline{or}^{1.5}$ russell $\underline{or}^{1.5}$ ada $\underline{or}^{1.5}$ algol $\underline{or}^{1.5}$ ell))

Sample Queries and Initial Boolean Formulations

Table 2

	Document 2703		Document 2932	
P-Value	Retrieval Rank	Query-Doc Similarity	Retrieval Rank	Query-Doc Similarity
p=∞	130	1.000	82	1.000
p=1	1	.091	12	.032
p=2	3	.207	4	.056
p=mixed	1	.146	5	.050

a) Output Ranks for I 2703 and I 2932

	Document 35	
P-Value	Retrieval Rank	Query-Document Similarity
p=∞	323	.0
p=1	138	.033
p=2	60	.075
p=mixed	53	.059

b) Output Ranks for I 35

	Document 2941		Document 2956	
P-Value	Retrieval Rank	Query-Doc Similarity	Retrieval Rank	Query-Doc Similarity
p=∞	1000	.0	1720	.0
p=1	15	.062	18	.069
p=2	89	.023	92	.023
p=mixed	20	.033	23	.032

c) Output Ranks for I 2941 and I 2956
(original query statement)

	Document 2941		Document 2956	
P-Value	Retrieval Rank	Query-Doc Similarity	Retrieval Rank	Query-Doc Similarity
p=∞	19	1.000	1726	.0
p=1	9	.047	10	.047
p=2	22	.019	23	.018
p=mixed	21	.030	22	.030

d) Output Ranks for I 2941 and I 2956
(reformulated query)

Output Ranks for Selected Sample Documents

Table 3

Document Number	Retrieval Rank	Query-Doc Similarity	Document Number	Retrieval Rank	Query-Doc Similarity
Conventional Boolean (p=∞)			p=2		
2035	1	1.000	2035	2	.280
-	-	-	1502	3	.268
2501	552	0	2399	5	.248
1307	564	0	756	7	.248
1502	990	0	2299	10	.248
756	1647	0	2501	25	.136
2299	1714	0	1810	31	.136
1810	1191	0	1307	100	.015
2399	3037	0			

Retrieval Rank of Relevant Documents for Query 5

Table 4

Appendix A: Sample Documents

.I 2932
.T
Complexity of Computations
.W
The framework for research in the theory of complexity of computations is described, emphasizing the interrelation between seemingly diverse problems and methods. Illustrative examples of practical and theoretical significance are given. Directions for new research are discussed.
.B
CACM September, 1977
.A
Rabin, M. O.
.K
complexity of computations, algebraic complexity, in tractable problems, probabilistic algorithms
.C
5.25

a) Sample Document 2932 (relevant to query CACM 21)

.I 2703
.T
The Intrinsically Exponential Complexity of the Circularity Problem for Attribute Grammars
.W
Attribute grammars are an extension of context-free grammars devised by Knuth as a mechanism for including the semantics of a context-free language with the syntax of the language. The circularity problem for a grammar is to determine whether the semantics for all possible sentences (programs) in fact will be well defined. It is proved that this problem is, in general, computationally intractable.
Specifically, it is shown that any deterministic algorithm which solves the problem must for infinitely many cases use an exponential amount of time. An improved version of Knuth's circularity testing algorithm is also given, which actually solves the problem within exponential time.
.B
CACM December, 1975
.A
Jazayeri, M.
Ogden, W. F.
Rounds, W. C.
.K
attribute grammars, circularity problem, context-free grammars, computational complexity, exponential time, semantics
.C
4.12, 5.25

b) Sample Document 2703 (relevant to query CACM 21)

.I 35
.T
Comparisons of Four Types of Lexical Indicators of Content
.A
Rath, G. J.
Resnick, A.
Savage, T. R.
.W
An experiment was conducted to determine which of four types of lexical indicators of content could be utilized best by subjects to determine relevant from irrelevant documents and to answer a set of 100 questions. The results indicate that there were no major differences between the groups using complete text and abstracts to select relevant documents, but the group utilizing the complete text obtained a significantly higher score on the examination.

c) Sample Document 35 (relevant to query CISI 1)

.I 2941
.T.
Early Experience with Mesa
.W
The experiences of Mesa's first users-primarily its implementers-are discussed, and some implications for Mesa and similar programming languages are suggested. The specific topics addressed are: module structure and its use in defining abstractions, data-structuring facilities in Mesa, an equivalence algorithm for types and type coercions, the benefits of the type system and why it is breached occasionally, and the difficulty of making the treatment of variant records safe.
.B
CACM August, 1977
.A
Geschke, C. M.
Morris, J. H. Jr.
Satterthwaite, E. H.
.K
programming languages, types, modules, data structures, systems programming
.C
4.22

d) Sample Document 2941 (relevant to query CACM 40)

.I 2956
.T
Some Ideas on Data Types in High-Level Languages
.W
A number of issues are explored concerning the notion that a data type is a set of values together with a set of primitive operations on those values. Among these are the need for a notation for iterating over the elements of any

finite set (instead of the more narrow for i := 1 to n notation), the use of the domain of an array as a data type, the need for a simple notation for allowing types of parameters to be themselves parameters (but in a restrictive fashion), and resulting problems with conversion of values from one type to another.
.B
CACM June, 1977
.A
Gries, D.
Gehani, N.
.K
data types, generic procedures, programming languages
.C
4.12, 4.20, 4.22

e) Sample Document 2956 (relevant to query CACM 40)

.I 1502
.T
An Online Editor
.W
An online, interactive system for text editing is described in detail, with remarks on the theoretical and experimental justification for its form.
Emphasis throughout the system is on providing maximum convenience and power for the user. Notable features are its ability to handle any piece of text, the content-searching facility, and the character-by-character editing operations. The editor can be programmed to a limited extent.
.B
CACM December, 1967
.A
Deutsch, L. P.
Lampson, B. W.

f) Sample Document 1502 (relevant to query CACM 15)