Approaches to Text Retrieval for Structured Documents

Gerard Salton*
Chris Buckley*

TR 90-1083
January 1990

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*Department of Computer Science, Cornell University, Ithaca, NY 14853. This study was supported in part by the National Science Foundation under grant IRI-87-02735.

Approaches to Text Retrieval for Structured Documents

Gerard Salton and Chris Buckley*

January 11, 1990

Abstract

Documents such as textbooks, dictionaries, and encyclopedias are inherently structured, in the sense that they are meant to be used selectively by skipping from section to section instead of reading sequentially from one end to the other. Experiments are described designed to provide selective reading lists for textbook materials in answer to questions submitted by the user population. A textbook in information science is used for experimental purposes.

1 Structured Text Collections

It is well known that collections of written text are inherently structured. For example, explicit text relationship indicators are often provided in the form of cross-references, footnotes and citations, and implicit content relationships exist between the sentences in a given paragraph, and between different paragraphs, and different documents.

Such text relationships have been used in the past in various ways — for example, in collection clustering, and relevance feedback. Clustering techniques are designed to group documents into affinity classes, making it possible to carry out efficient collection searches and to retrieve classes of similar items in a single search operation [1-4]. Analogously, relevance feedback is used to improve search statements, and hence to retrieve new relevant items, by utilizing relevance assessments obtained from system users for previously retrieved documents [5-7].

The recent work in the *hypertext* area also uses text structure to simplify text traversal and text retrieval operations [8-10]. In that case, links are placed

*Department of Computer Science, Cornell University, Ithaca, NY 14853. This study was supported in part by the National Science Foundation under grant IRI 87-02735.

between related pieces of text, and these links are followed during retrieval to identify related texts, or text excerpts. Structured text representations are especially useful for texts that are not meant to be read sequentially from one end to the other. The links are then used to provide selective retrieval of certain linked text portions.

In an information retrieval context, several text-structuring problems must be faced:

- How to subdivide the texts into linking units that provide advantages in information retrieval.
- How to identify relatable text portions while supplying the corresponding text links.
- How to traverse a linked text for retrieval purposes.
- How to measure the retrieval effectiveness in a structured text environment, compared with the retrieval in ordinary text collections.

These questions are examined in the remainder of this study.

2 Automatic Text Structuring

In some environments, the basic text structuring task is largely self-defined. For example, when dictionaries, thesauruses, or encyclopedias are used as a retrieval base, the linking unit almost surely consists of individual entries or encyclopedia units. In that case the library system is designed to facilitate jumping from one dictionary entry or one encyclopedia article to a related one.

When more general texts are processed, appropriate linking units might be defined before structured text searches are possible. In the search operations implemented for the complete texts of Shakespeare on the NeXT machine, the basic retrieval unit was chosen either as one complete Shakespeare sonnet (14 lines), or else one scene of a Shakespeare play — typically 100 to 200 lines of text. These individual text units constitute *local documents* that are treated as separate text units within the context of the larger document collection.

When conventional running texts are available in a retrieval environment, such as complete books or journal articles, a whole book chapter may cover many pages of text and deal with a variety of different topics. An individual text sentence, on the other hand, is often very confined and difficult to interpret out-of-context. In the experiments conducted in this study, complete *paragraphs* of text are used as linking units, the assumption being that the content within a paragraph is sufficiently homogeneous to be used as a basic unit for retrieval purposes.

When text paragraphs are treated as retrieval units, the content linking task can then be handled in the following way:

a) Individual text paragraphs are recognized.

b) An indexing system is used to identify paragraph content and to assign content terms.

c) The paragraph descriptions are compared and links are supplied between paragraphs with sufficiently high content similarily.

The important step is the paragraph content identification. Over the last few decades, viable *automatic indexing* systems have been developed that are capable of assigning to each text item a set of important terms used for content identification. Given a text item D_i (a particular text paragraph), the text content is often represented as a set, or vector, of terms $D_i = (d_{i1}, d_{i2}, ..., d_{it})$ where d_{ik} represents an importance factor, or *weight*, of term T_k assigned to item D_i [6, 7, 11, 12].

A high performance term weighting system normally takes into account the frequency with which a term is used in a particular document, the number of documents in a collection to which a term is assigned, and the document length or number of terms occurring in a document. The so-called *tf* × *idf* (term frequency times inverse document frequency) strategy assigns high term weights to text elements that occur frequently inside a particular document but relatively rarely in the collection as a whole. Terms with a large *tf* × *idf* factors are know to be important for content indentification purposes. [13-15]

Given two paragraphs D_i and D_j both represented by term vectors, a similarity measure may be computed between the two items based on the number and the weight of jointly assigned terms. Mathematically, the similarity between two text items $D_i = (d_{i1}, d_{i2}, ..., d_{it})$ and $D_j = (d_{j1}, d_{j2}, ..., d_{jt})$ can be measured by the inner product between the corresponding term vectors as follows:

$$Sim(D_i, D_j) = \sum_{k=1}^{t} d_{ik} \cdot d_{jk} \tag{1}$$

where t is the total number of assignable content terms. When *tf* × *idf* weights are used to reflect term importance and the similarity computations are normalized for document length, the pairwise similarity Sim (D_i, D_j) produces values between 0 and 1.

Global similarity computations such as those of expression (1) are usable for document classification when classes of items are defined consisting of items exhibiting a sufficiently high pairwise global similarity. For text classification purposes pairwise similarity measures must be computed between paragraph pairs and grouping criteria must be defined to generate classes of mutually related text items. Typically, hierarchical text classification systems can be constructed by first forming small groups of highly related items (where each group consists of a small number of items with a large pairwise similarity). The small tighly related classes may be expanded into larger groupings with a smaller overall similarity. When this process is continued, one large heterogeneous class is formed at the end consisting of all items in the text collection. [1-4]

Fig. 1 shows an excerpt of a cluster (class) hierarchy constructed for the paragraphs of a textbook in information science.[16] In the illustration of Fig. 1, three low-level clusters are defined consisting of items (397 and 791), (642 and 644), and (655 and 656). The global similarity coefficients obtained for the respective term vectors (see expression (1)) are included in the figure ranging from 0.605 for items 642-644 to 0.556 for 397-791. The two groups consisting of (642, 644) and (655, 656) are themselves grouped into a larger class with an overall similarity of 0.410, implying that the smallest similarity between any pair in the group (642, 644, 656, and 656) is 0.410. Finally the group of 4 items is joined with another group of two items consisting of (397, 791), the global similarity of the complete set of 6 items being 0.329 (the smallest similarity between some element in group (397, 791) and some other element in (642, 646, 655, 656) is 0.329.

A hierarchical representation of the cluster of Fig. 1(a) is shown in Fig. 1(b), where the actual documents (paragraphs) are represented by the leaves of the tree, and the interior nodes specify the respective clustering similarity. In the illustration of Fig. 1(b), the four paragraphs of chapter 9 of [16] cover topics dealing with the generation of word stems, the assignment of content identifiers to the documents of a collection, and the general automatic indexing process. Item 397 from chapter 7 discusses text decomposition of words and affixes, and item 791 from chapter 11 deals with word morphology from a linguistic viewpoint. All of these topics are related to word stemming and automatic indexing.

In principle, a hierarchival document or paragraph classification can be used directly to define an appropriate linking structure usable for text retrieval:

a) An incoming query may be compared with all existing paragraph descriptions.

b) The best matching text paragraphs can be retrieved (say paragraph 642 in the illustration of Fig. 1).

c) Additional paragraphs are retrieved from the same cluster, assuming that the reader wishes to see more output materials.

d) The search may be expanded to adjacent clusters of items (say 655, 656), if the user wishes to obtain still more information.

When the paragraph similarities are sufficiently high – say above 0.400 on a similarity scale ranging from 0 to 1 – properly related paragraph sets may emerge with such a cluster search process.

In practice, as the clustering example of Fig. 1 shows, jumping from cluster to adjacent cluster often uses links of low similarity, possibly indluding large-scale topic changes. A greater degree of confidence in the appropriateness of the global paragraph linking mechanism may be gained by constructing chains of mutually similar items, where item A is closely related to item B, which is in turn closely linked to C, and so on. An iterated similarity computation system may then be used to construct linked paragraph chains, starting with one or more seed items known to be relevant to the user:

a) Each seed item is compared with all other text items in the collection, and a similarity threshold is used to identify one or more related items.

b) The related items are used next as seed items and the search process is iterated to produce still more related items; the similarity threshold may be varied form one iteration to the next to control the number of related items retrieved in each iteration.

In the foregoing discussion, all paragraph comparisons are assumed to be carried out globally, by comparing the term vectors attached to the respective paragraphs using the model of equation (1). The likelihood of useful paragraph links may be increased in some circumstances by comparing *sentences* in highly matching paragraph pairs, and retrieving a linked item only if the global similarity with a seed item exceeds a stated treshold, *and* if at least one (or more) matching sentence pairs are found in the respective paragraph pair. Substantial evidence exists that the presence of highly matching sentences in pairs of paragraphs provides evidence of content relationship between text excerpts. A pairwise sentence comparison may then be performed optionally for paragraphs with high global similarity, in addition to the global paragraph comparisons.

When sentences are compared, a global similarity based on normalized term weights, such as those of expression (1), may not be suitable. For short sentences involving only one or two significant terms, the global similarity with normalized weights will produce perfect similarity coefficients of 1 for many sentence pairs. Furthermore, the inverse document frequency (*idf*) factor that depends on the number of documents in which a term occurs is not unambiguously defined in a sentence context.

For sentence similarity computations, it then appears preferable to use as a weight for term k in sentence S_i, the term frequency tf_{ik}, representing the number of occurences of term k in S_i. In addition, an extra weight might be attached to matching sequences of significant terms that occur adjacently in the respective sentences, or that occur in close proximity of each other in the sentences. For purposes of this study, the similarity between sentences S_i and S_j is obtained simply as the sum of the minimum term frequency weights of matching terms in the sentences S_i and S_j:

$$Sim(S_i, S_j) = \sum_{\substack{\text{matching}\\ \text{terms } k}} min(tf_{ik}, tf_{jk}) \tag{2}$$

Consider as an example documents 1032 and 1035 reproduced in Fig. 2. The sentences of documents 1032 and 1035 are numbered from 00 to 05, and 00 to 04 respectively. Following deletion of common function words entered on a word exclusion list, and reduction of the remaining text words to word stem form, each text sentence is represented by a set of significant word stems, as shown in Fig. 2(c) for sentence 02 of document 1032 (labeled 103202) and sentence 04 of document 1035 (103504). The similarity score between the sentences is based on the number of common terms in the sentences, and the occurrence frequency of the common terms. For sentences 103202 and 103504, the following common set of terms and frequency assignments are obtained:

$S_{1032-02}$: (discard (1), incom (1), mail (2), mess (3))
$S_{1035-04}$: (discard (1), incom (1), mail (1), mess (4))

The similarity formula (2) thus produces a matching coefficient of $1 + 1 + 1 + 3 = 6$ for the two sentences.

In the sentence matching procedure, extra weight might be given to matching sequences of common words that occur adjacently in the sentence texts. For example, if the phrase "incoming mail" had occurred jointly in the two sample sentences, a matching weight would be computed for "incoming" and for "mail". In addition, a further *phrase weight* might be added for the number of matches between the complete phrase "incoming mail".

In the experiments which follow, a variable similarity threshold is used for both global document and sentence matches, designed to insure that the number of new retrieved documents in a given iteration is not smaller than the number produced in the previous iteration.

3 Retrieval of Structured Text Elements

The retrieval experiments described in this study are based on the analysis of the complete text of "Automatic Text Processing".[16] This text is divided into 1,140 paragraphs (local documents) and about 4,500 sentences. The relevant statistics are summarized in Table 1. Ten sample queries are used, each corresponding to section headings included in the text of reference [16]. The query texts are compared in each case with the indexed representations of all 1140 document texts, and an iterated process is used to retrieve the best (most highly matching) paragraphs from the textbook. The process used for the document and sentence comparison is outlined in Table 2.

As the table shows, two main procedures are used. In the first one, consisting of steps 1, 2, 3a, and 4 of Table 2, document-document matches are used to retrieve items whose global similarity with some previously available document exceeds a stated threshold. In the second process, consisting of steps 1, 2, 3b and 4 of Table 2, a global document-document match does not lead to immediate retrieval, but new documents are retrieved only if the sentence match between a sentence in a new document and a sentence in a previously available item exceeds a given threshold. In either case, that is, for both document-related and sentence-related output, variable thresholds in the global document similarity, or in the sentence similarity, are used to determine how many documents are to be retrieved at any time. For the present experiments, the threshold is picked in such a way that the number of new documents identified in each iteration exceeds the number of distinct old documents used in the previous iteration.

The retrieval output for query Q1 "Electronic Mail and Messages" is shown in Tables 3 and 4 for document-related and sentence-related output, respectively. As the Tables show, two documents that are most similar to the query statement are retrieval in the initial pass. These documents then serve as seeds for a global comparison with all other documents in pass 1. The retrieval threshold is set at 0.35 for the global document similarity (equation (1)) that controls the document-related output, and the sentence similarity it fixed at 6 for the sentence-related output. Such a threshold setting produces 5 new items, in pass 1 for both processes.

The newly found items from pass 1 are in turn used as seeds for pass 2 with thresholds 0.40 and 8, respectively, producing 6 distinct new items for both processes. Pass 3 is carried out with threshold at 0.35 and 7, producing 6 and 10 new items for document- and sentence-related outputs, respectively.

The global retrieval results for query 1 are summarized in the output of Table 5. The retrieved sets for the two procedures have 13 items in common. Six additional items are obtained only through the document-related output, and 10 more items are produced only by the sentence-related output. The two seed documents (1032 and 1043) represent paragraphs in chapter 13 of [16] covering the topic of electronic mail and messages. A large number of additional documents from chapter 13 are retrieved in pass 1, together with one item from chapter 3 dealing with office automation and the use of electronic mail in offices. In pass 2, the search broadens considerably to include several items from chapter 2 dealing with computer hardware and network design, plus the additional item from chapter 5 dealing with statistical language analysis and message entropy computations. Finally in pass 3, more items are retrieved from chapters 2, 3, 5 and 13, and an additional item from chapter 6 dealing with cryptography and message enciphering.

For each query, retrieval maps can be produced such as those shown in Fig. 3 for the document-related and sentence-related outputs of query 1. Such maps can help system users to control the retrieved output. Conservative users who wish to receive a thorough introduction to a topic may wish to utilize breadth-first searches covering many documents on the same level of the search tree. More adventurous users may rapidly jump from one tree level to another by using depth-first approaches covering documents from many different book chapters. Typical breadth-first and depth-first search strategies using the document-related output of Fig. 3(a) are shown in Table 6.

The search of Table 6(a) covers the initial documents in detail as well as all other items recovered in pass 1. For subsequent passes, only those documents are used that originate in chapters not previously seen. The depth-first search of Table 6(b) picks a particular top-down search path that proceeds directly from the upper levels to the lower levels of the search tree.

Retrieval maps such as those of Fig. 3 can also help in placing content links between related paragraphs, either fully automatically, or semi-manually under author control. In the latter case, document texts such as those in Fig. 2 can be displayed selectively to help the author in the link placement.

Table 7 presents as overall evaluation of the paragraph retrieval system carried out with 10 sample queries used with the text of reference. [16] For each query, the table shows the number of documents retrieved in each pass by the variable threshold method for the document-related and sentence-related processes. About 30 documents (paragraphs) are obtained on average for each query. The performance is flawless (A rating) for 2 queries out of 10, and quite acceptable for 5 more queries (B rating). One query received a C (questionable) rating, and 2 more queries have a D (poor) rating.

In general, the document-related output is more secure in retrieving useful items, and the search is generally more concentrated in the basic chapters in the document-related process. The sentence-related output roams further afield and is more varied, because documents with somewhat lower global pairwise similarity are reached when a sentence match is required for retrieval. For 4 queries out of 10 (queries 2, 5, 8, 10), the document-related output is preferred. The two types of output are equivalent for four more queries (1, 3, 6, 9). For the two remaining queries (numbers 4 and 7), the sentence-related output is preferred, because the document-related output fails to produce an adequate number of new items in these cases. For query 4, only two retrieved items are specific to the document-related process, whereas for query 7 the document-related process reaches too many items (25) of which many are extraneous. The sentence-based process is thus useful when the document-related method fails.

The structured text retrieval method described in this note remains to be evaluated in a user environment with actual user queries.

REFERENCES

1. N. Jardine and C. J. van Rijsbergen, The Use of Hierarchic Clustering in Information Retrieval, *Information Storage and Retrieval*, 7:5, December 1971, 217-240.

2. G. Salton and A. Wong, Generation and Search of Clustered Files, *ACM Transactions on Database Systems*, 3:4, December 1978, 321-346.

3. F. Murtagh, A Survey of Recent Advances in Hierarchical Clustering Algorithms, *The Computer Journal*, 26:4, 1982, 354-360.

4. P. Willett, A Fast Procedure for the Calculation of Similarity Coefficients in Automatic Classification, *Information Processing and Management*, 17:2, 1981, 53-60.

5. J.J. Rocchio Jr., Relevance Feedback in Information Retrieval, in *The Smart System - Experiments in Automatic Document Processing*, G. Salton, editor, Prentice Hall Inc., Englewood Cliffs, NJ, 1971, 313-323.

6. C. J. van Rijsbergen, *Information Retrieval*, Butterworths, London, Second Edition, 1979.

7. G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw Hill Book Co., New York, 1983.

8. W. B. Croft and H. Turtle, A Retrieval Model for Incorporating Hypertext Links, *Proceedings Hypertext 89*, Pittsburgh, PA, November 1989, 213-224.

9. M. E. Frisse, Searching for Information in a Hypertext Medical Handbook, *Communications of the ACM*, 31:7, July 1988, 880-886.

10. J. Conklin, Hypertext: An Introduction and Survey, *Computer*, 20:9, September 1987, 17-41.

11. G. Salton, A Theory of Indexing, *Regional Conference Series in Applied Mathematics No. 18*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1975.

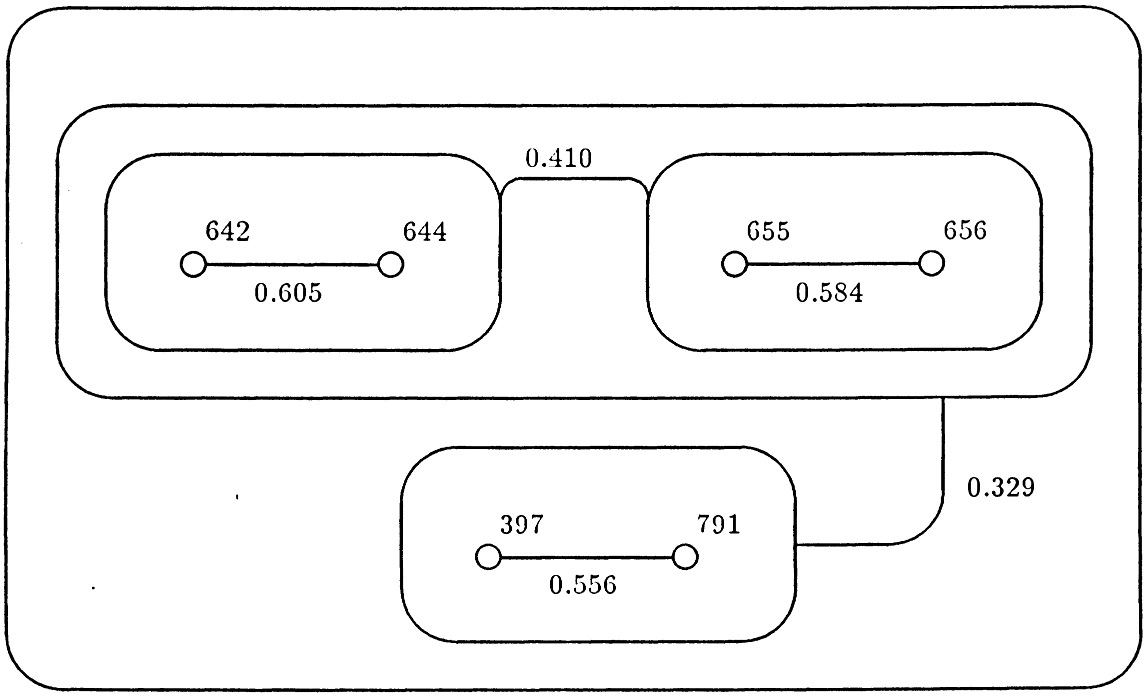
12. G. Salton, A Blueprint for Automatic Indexing, *ACM SIGIR Forum*, 16:2, Fall 1981, 22-38.

13. G. Salton and C. S. Yang, On the Specification of Term Values in Automatic Indexing, *Journal of Documentation*, 29:4, December 1973, 351-372.

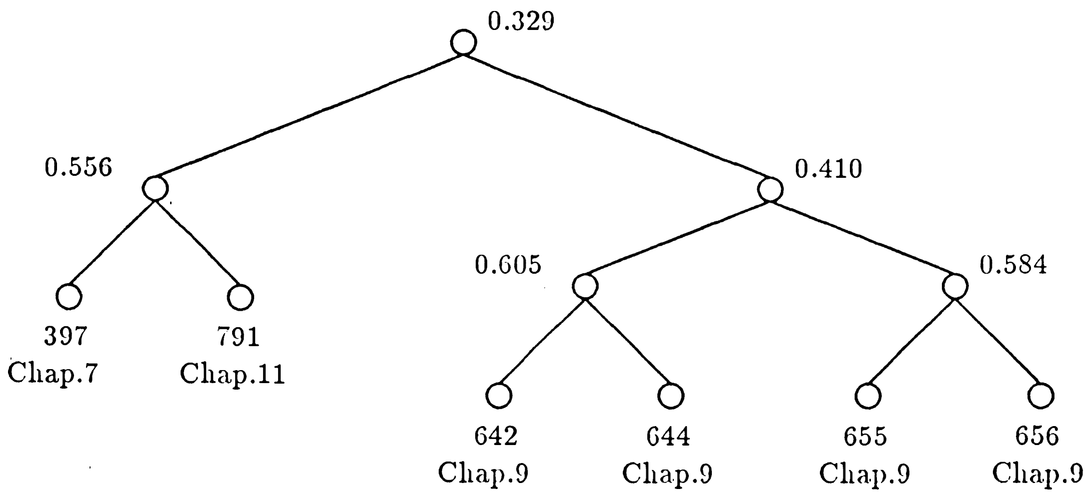
14. G. Salton, C. S. Yang and C. T. Yu, A Theory of Term Importance in Automatic Text Analysis, *Journal of the ASIS*, 26:1, January-February 1975, 33-44.

15. G. Salton and C. Buckley, Term Weighting Approaches in Automatic Text Retrieval, *Information Processing and Management*, 24:5, 1988, 513-523.

16. G. Salton, *Automatic Text Processing*, Addison Wesley Publishing Company, Reading MA, 1989.



a) Typical Cluster of Text Paragraphs



b) Hierarchival Cluster Representation

Figure 1: Paragraph Clustering

.I 1032

00 13.6 Electronic Mail and Messages

01 An electronic mail facility is a communications system that allows participants to send each other mail and messages using electronic methods of information transmission.

02 Among the main features of such systems are provisions for entering text *messages*, mailing *messages*, informing the intended recipients of the arrival of *messages* and allowing the recipients to read, file or *discard* incoming *mail.*

03 Electronic-messaging systems are popular among many users because they simplify the composition and transmission of messages, while also increasing the transmission speed and reducing the cost of communication.

04 Automatic mail systems may also increase users' productivity– the sender avoids the inconvenience of dealing with busy telephone lines and unanswered phones.

05 Furthermore, since mail can be forwarded and received at any time, electronic mail-handling systems need not interrupt other activities.

a) Sample Document 1032

.I 1035

00 A useful automatic message processing system includes the following components: [52,53]

01 An interface program between the user's applications programs and the communications system that actually transmits information.

02 The interface system should save incoming messages temporarily until the user's applications program is ready to take over, and should maintain queues of outgoing messages ready for transmission but not yet delivered to the network.

03 A message-editing system that packs messages into segments, and interprets the destination and routing information in the message headers.

04 A mailbox service that classifies *messages* in priority order, lists *incoming messages*, counts and inspects mailbox contents, stored particular *messages*, handles *mail* inquiries, releases messages, and eliminates items to be discarded.

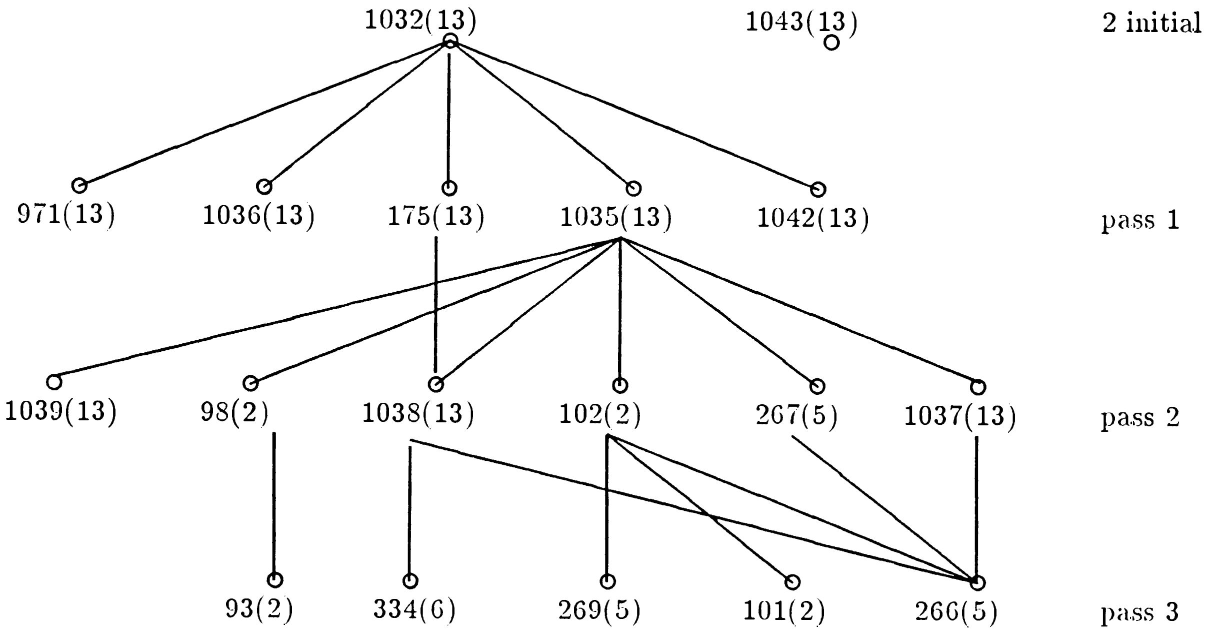
b) Sample Document 1035

103202 main featur system provis enter text *mess mail mess* inform intend recipi arriv *mess* allow recipi read file *discard incom mail*

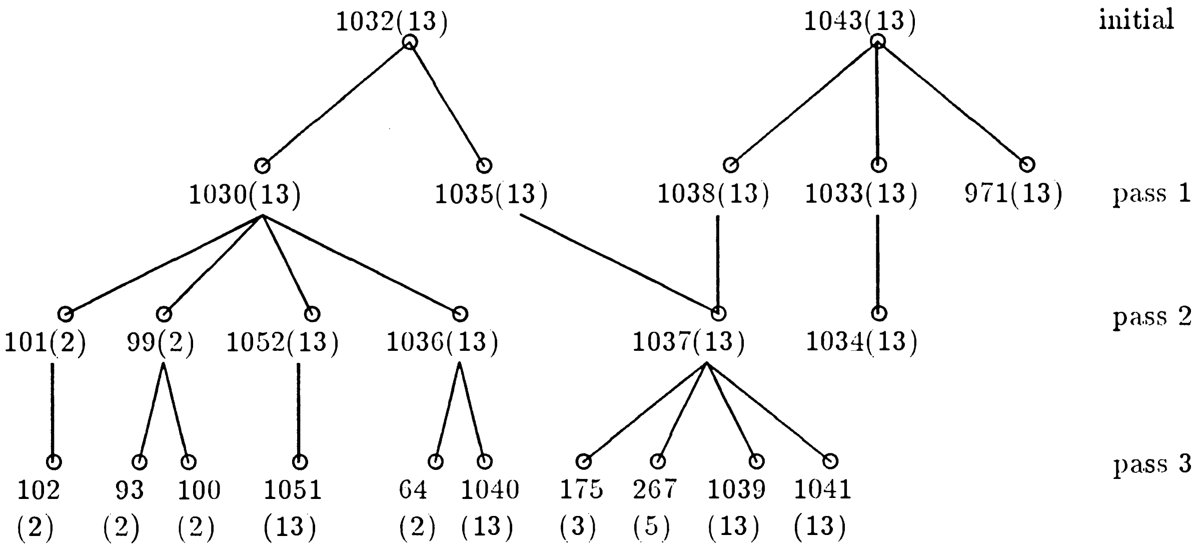
103504 mailbox servic classif *mess* prior ord list *incom mess* count inspect mailbox content stor *mess* handl *mail* inquir releas *mess* elim item *discard*

c) Significant Terms for 1032-02 and 1035-04

Figure 2: Examples of Sentence Indexing



a) Document-Related Retrieval Map for Query 1



b) Sentence-Related Retrieval Map for Query 1

Figure 3: Retrieval Map for Query 1

Number of distinct documents (paragraphs)	1,140
Number of distinct document (paragraph) pairs	about 650,000
Global similarity coefficient for top 500 paragraph pairs (min sim = 0, max sim = 1)	0.886 to 0.526
Number of distinct text sentences	about 4,500
Number of distinct sentence pairs	about 10,125,000
Similarity coefficient for top 500 sentence pairs (min sim = 0)	19 to 8

Table 1: Document and Sentence Statistics for Text of Reference [19]

Step 1: For each query text, retrieve the best two documents (topdoc) using global term vector comparison with normalized (tf x idf) concepts.

Step 2: Compare each topdoc with the remaining documents and arrange the ten best items for each topdoc in decreasing global document-document similarity order. Use this list in step 3.

Step 3a) Document-related process: determine the global similarity threshold (0.25, 0.35, 0.45,...) that retrieves more new documents from the list of step 2 than the current number of distinct topdocs. The newly found items with global similarity exceeding the threshold form the new topdoc items.

Step 3b) Sentence-related process: for each pair of documents consisting of one topdoc and one item from list of items obtained in step 2, determine all common terms for the dominent pair. Index the sentences of the document pair with all common terms plus common phrases formed by adjacent common terms. Determine the sentence matching threshold (4,5,6,...) that identifies more new documents from the list of step 2 than the current number of distinct topdocs. All newly identified documents with at least one matching sentence with a topdoc form the new set of topdocs.

Step 4: Repeat steps 2 and 3 for three iterations.

Table 2: Document and Sentence Matching Process

Query 1 Electronic Mail and Messages

Pass 0:	Initial 2 topdocs	1032 (chapter 13)	
		1043 (chapter 13)	
Pass 1:	Match with initial topdocs	0.35 threshold	
	1032 + 175 (chapter 3)	sim 0.45	
	1032 + 971 (chapter 13)	sim 0.35	5 new
chapter 3,13	1032 + 1035 (chapter 13)	sim 0.38	retrieved
	1032 + 1036 (chapter 13)	sim 0.35	items
	1032 + 1042 (chapter 13)	sim 0.37	
Pass 2:	Match with new topdocs	0.40 threshold	
	175 + 1038 (chapter 13)	sim 0.40	
	1035 + 98 (chapter 12)	sim 0.41	7 new
chapters 2,5,13	1035 + 102 (chapter 2)	sim 0.46	retrieved
	1035 + 267 (chapter 5)	sim 0.45	(6 distinct)
	1035 + 1037 (chapter 13)	sim 0.61	
	1035 + 1038 (chapter 13)	sim 0.60	
	1035 + 1039 (chapter 13)	sim 0.41	
Pass 3:	Match with new topdocs	0.35 threshold	
	98 + 93 (chapter 2)	sim 0.36	
	102 + 97 (chapter 2)	sim 0.36	
	102 + 101 (chapter 2)	sim 0.36	
	102 + 266 (chapter 5)	sim 0.36	10 new
chapters 2,5,6	267 + 266 (chapter 5)	sim 0.68	retrieved
	1037 + 266 (chapter 5)	sim 0.43	(6 distinct)
	1038 + 266 (chapter 5)	sim 0.49	
	267 + 285 (chapter 5)	sim 0.78	
	267 + 334 (chapter 6)	sim 0.36	
	1038 + 334 (chapter 6)	sim 0.36	

Table 3: Document-Related Retrieval Process for Query 1

Query 1 Electronic Mail and Messages

Pass 0:	Initial 2 topdocs	1032 (Chapter 13)	
		1043 (chapter 13)	
Pass 1:	Match with initial topdocs	(sentence matching threshold 6)	
	1032 + 1030 (chapter 13)	sentence sim 6	
	1032 + 1035 (chapter 13)	sentence sim 6	5 new
chapter 13	1043 + 971 (chapter 13)	sentence sim 6	retrieved
	1043 + 1037 (chapter 13)	sentence sim 6	items
	1043 + 1038 (chapter 13)	sentence sim 6	
Pass 2:	Match with new topdocs	(sentence matching threshold 8)	
	1030 + 99 (chapter 2)	sentence sim 8	
	1030 + 101 (chapter 2)	sentence sim 8	7 new
chapters 2,13	1030 + 1036 (chapter 13)	sentence sim 9	retrieved
	1030 + 1052 (chapter 13)	sentence sim 11	items
	1033 + 1034 (chapter 13)	sentence sim 8	(6 distinct)
	1035 + 1037 (chapter 13)	sentence sim 8	
	1038 + 1037 (chapter 13)	sentence sim 12	
Pass 3:	Match with new topdocs	(sentence matching threshold 7)	
	99 + 93 (chapter 2)	sentence sim 7	
	99 + 100 (chapter 2)	sentence sim 7	
	101 + 102 (chapter 2)	sentence sim 7	
	1036 + 64 (chapter 2)	sentence sim 8	10 new
	1036 + 1040 (chapter 13)	sentence sim 7	retrieved
chapters 2,3,5,13	1037 + 175 (chapter 3)	sentence sim 8	items
	1037 + 267 (chapter 5)	sentence sim 7	(6 distinct)
	1037 + 1039 (chapter 13)	sentence sim 7	
	1039 + 1041 (chapter 13)	sentence sim 7	
	1052 + 1051 (chapter 13)	sentence sim 7	

Table 4: Sentence-Related Retrieval Process for Query 1

1. Retrieval by both document-related (D) and sentence-related (S) processes

13 items:	93(chap 2)	pass 3 D,S
	101 (chap 2)	pass 2S, 3D
	102 (chap 2)	pass 2D, 3S
	175 (chap 3)	pass 1D, 3S
	267 (chap 5)	pass 2D, 3S
	971 (chap 13)	pass 1 D,S
	1032 (chap 13)	initial item
	1035 (chap 13)	pass 1 D,S
	1036 (chap 13)	pass 1D, 2S
	1034 (chap 13)	pass 2 D,S
	1038 (chap 13)	pass 1S, 2D
	1039 (chap 13)	pass 2D, 3S
	1043 (chap 13)	initial item

2. Retrieved only by document-related output: 6 items

97 (chap 2) pass 3	98 (chap 2) pass 2	266 (chap 5) pass 3
269 (chap 5) pass 3	334 (chap 6) pass 3	1042(chap 13) pass 1.

3. Retrieved only by sentence-related output: 10 items

64 (chap 2) pass 3	99 (chap 2) pass 2	100 (chap 2) pass 3
1030 (chap 13) pass 1	1033 (chap 13) pass 1	1034 (chap 13) pass 2
1040 (chap 13) pass 3	1041 (chap 13) pass 3	1057 (chap 13) pass 3
1052 (chap 13) pass 2.		

Table 5: Global Retrieval Results for Query 1

1032 (13)	electronic mail and messages	(initial)
1043 (13)	electronic mail characteristics	
175 (3)	mail forwarding in offices	(pass 1)
971 (13)	general electronic information systems	
1035 (13)	message processing	
1036 (13)	inplementation of mail and messages	
1042 (13)	confidentiality of messages	
98 (2)	routing methods for message	(pass 2)
102 (2)	message switching in networks	
207 (5)	message entropy	
334 (6)	message encrypting	(pass 3)

a) Sample Breadth-First Search

1032 (13)	electronic mail and messages
175 (3)	mail forwarding in offices
1038 (13)	message transfer systems
334 (6)	message encrypting
266 (5)	network architecture

b) Sample Depth-First Search

Table 6: Sample Search for Retrieval Map of Fig. 3(a).

Query	Text	Number of Retrieved Items Common Items	 Document Related	 Sentence Related	Rating
1.	Electronic Mail and Messages (D,S output equivalent)	13	6	10	A
2.	Approaches to Text Generation (D output preferable; S output very varied)	7	10	16	C
3.	Extended Boolean System (D,S output equivalent)	9	9	11	A
4.	Multidimensional Access Structures (D output very restricted; S output preferable)	13	2	11	B
5.	Relational Database System (D output adds relevant items; S output too varried)	4	15	13	B
6.	Special Purpose Compression Systems (D,S outputs equivalent)	11	6	5	B
7.	Ciphers based on computationally hard problems (D output picks up too much; S output is preferred)	5	25	10	B-
8.	Interactive Graphic Editing System (D output more concentrated, S output too varied)	6	12	12	B-
9.	Data Security, Integrity, Recovery (D,S outputs equivalent)	11	6	9	D
10.	Natural Languages Interface to Information Systems (D output adds good items; S output very varied)	10	14	14	D
	Average	8.9	10.6	11.1	

Table 7: Global Rating of Retrieval Performance