Flexible Text Matching for Information Retrieval

Gerard Salton
Chris Buckley

TR 90-1158
September 1990

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

*Department of Computer Science, Cornell University, Ithaca, NY 14853-7501. This study was supported in part by the National Science Foundation under grant IRI-87-02735.

Flexible Text Matching for Information Retrieval

Gerard Salton and Chris Buckley *

Abstract

Very large text databases now exist in machine-readable form, covering arbitrary subject matter in unrestricted discourse areas. The conventional text retrieval approaches are not easily used in such circumstances, because the knowledge needed to understand unrestricted subject matter is not readily available for practical use.

A new approach is outlined for text structuring and retrieval, based on flexible text matching methods using different context granularities. When global as well as local similarities exist between distinct texts, the presumption is that the texts cover semantically similar subject areas. This leads to the automatic introduction of links between related texts, and to the retrieval of text excerpts in response to available user queries. Evaluation results are given to demonstrate the effectiveness of the text matching approach.

1 The Text Retrieval Problem

A serious information problem has existed around the world for many years, because information needed to make decisions, or to carry out particular tasks, is often unknown or unavailable. The problem has grown worse in recent times as the amount and variety of potentially available information have increased, and the search and retrieval tasks needed to gain access to particular information items have become more complex.[1-3] To be sure, noticeable changes have occurred in the physical environment in which information retrieval activities take place, exemplified by the introduction of high-density storage devices capable of storing hundreds of millions of bytes of information, the use of sophisticated information display techniques, and the construction of communications networks capable of putting the user in touch with many information resources. These

*Department of Computer Science, Cornell University, Ithaca, NY 14853-7501. This study was supported in part by the National Science Foundation under grant IRI 87 02735

software and hardware advances in the retrieval environment have not unfortunately been accompanied by corresponding improvements in the intellectual tools used to gain access to the stored data.

It was hard enough in earlier years to deal with homogeneous collections of documents where all items exhibit the same length and cover related subject areas and linguistic contexts. Nowadays, the widespread availability of machine-readable data makes it necessary to deal with large collections of heterogeneous materials: letters, messages, articles, books, dictionaries, encyclopedias, and so on. Because the appearance of these items varies enormously and the subject areas are not tightly controllable, new approaches are needed for information handling. The following tasks are especially important:

- the content analysis and content representation of information items included in heterogeneous collections;
- the structuring of heterogeneous data collections to simplify data access – for example by forming groups of related items, or by linking items exhibiting sufficiently large subject similarities;
- the rapid comparison of query and document representations, and the identification of stored items with sufficiently large query similarities;
- the design of flexible systems for interaction between the users and the information system.

Many of these operations are related, and all of them ultimately depend on the methods used to represent text content. Some additional background is therefore provided in the content analysis area.

2 Approaches to Text Analysis

In thinking about the content representation of natural-language texts, two somewhat extreme viewpoints may be distinguished:

- A shallow keyword representation may be used where individual words, sometimes also known as terms or keywords, are assigned to the information items, to be used collectively for the representation of document content. When the text words, or derivations of the text words, are used as keywords, one speaks of a full-text information system. In keyword systems, the retrieval decisions are based on keyword comparisons, followed by the retrieval of items containing keyword patterns corresponding to those used in the query specifications.
- A deep knowledge-based representation is sometimes proposed where a formal knowledge structure is first constructed for a particular topic area,

containing the main entities and relationships between entities of interest in that subject domain. The content of each document is then specified by certain formal constructs related to the available formal knowledge representation. When knowledge-based approaches are used, retrieval of particular items depends on comparisons between formal constructs attached to documents and information requests, and on inference procedures used to derive the query representations from the available document representations, or vice-versa.

When large masses of heterogeneous texts must be processed, neither the shallow keyword approach, nor the deep knowledge-based approach are fully satisfactory for the content representation of texts. The text content can never be represented fully accurately by using sets of unrelated single terms or keywords, even when some control is imposed on the assignment of the terms, and weights are assigned to the terms in order of presumed term importance. In fact, critics have charged that the retrieval performance of keyword systems must remain mediocre when terms extracted mainly from document texts are used for retrieval:[4]

> "Full-text retrieval systems applied to large databases are unlikely to perform well in any retrieval environment because the user (who formulates the search request) cannot foresee the exact words and phrases ... used in the documents."

On the other hand, the deep semantic approaches are also not immediately viable in unrestricted text handling situations of the type that now exist in practice. Typically, knowledge bases providing semantic subject characterizations have been built for limited contexts – stories dealing with certain traffic accidents, or with some financial mergers and acquisitions, or with certain violent crimes. In these circumstances, limited text analysis operations can be carried out for documents that fit the semantic environment of the classified knowledge.[5-7] But those approaches are not extendable to cover wider contexts, because quite different knowledge structures may then be needed to characterize the items, including in particular a good deal of common sense knowledge in addition to the more narrow technical know-how used in restricted contexts.

Some intermediate approaches have been proposed for the text analysis problem in which a full semantic characterization is avoided, but limited relationships are recognized between the linguistic elements used to represent the written texts. Typically, term phrases rather than single terms may then be assigned for text content identification. Term phrases can be generated using statistical approaches based on the co-occurrence in particular sentence contexts of the various phrase components; alternatively, syntactic analysis techniques are usable for the recognition of noun phrase, or prepositional phrase, structures.[8-10] When auxiliary vocabulary schedules are available for use, certain semantic relationships between terms may be recognized, including especially synonymous term relations obtained from stored thesauruses [11,12], and hierarchical inclusion relations extracted from machine-readable dictionaries.[13,14]

Unfortunately, the available experience indicates that reliable and consistent improvements in retrieval effectiveness beyond the performance of properly weighted single-term keywords are difficult to obtain by using thesaurus and phrase assignments. The problem is that vocabulary schedules and thesauruses constructed for particular document collections are not easily applied to new situations and new collections. Hence a given thesaurus may provide only locally valid improvements that perform satisfactorily under special circumstances.

Phrase assignments are relatively easy to implement, but the necessary degree of discrimination needed to distinguish useful phrases from extraneous ones is normally lacking. Thus, a particular phrase formation process generates both good and bad phrases, and for this reason it has not been possible to obtain consistent improvements with any phrase assignment system that has been tried in practice. In general, the simpler text analysis methods including certain single-term indexing methods appear to hold up remarkably well in many different contexts, compared with other, more refined text content representations that help only in special circumstances.[15]

Consider, as an example, the results of Fagan using statistically and syntactically generated phrase assignments, in addition to a standard term assignment system based on sets of weighted, single terms.[16,17] The results of Table 1 cover two different text collections, including 3204 documents in computer science used with 52 search queries, and 1460 documents in library and information science used with 76 queries. In each case, one average value of the search precision is computed for each query for 10 values of the recall, and these average precision values are averaged in turn over the whole query set to obtain the final composite precision figure shown in Table 1.[1] As the Table shows, significant improvements in the composite precision are obtained for the statistical co-occurrence phrases for one of the two text collections (CACM), but hardly any difference in performance is noted for the other text collection. Furthermore, the performance of the syntactic phrases is disappointing for both collections.

To summarize, when dealing with heterogeneous text collections in arbitrary subject areas, fully refined linguistic analysis methods cannot be carried out, because the full-blown knowledge structures that specify the needed seman-

[1] The retrieval precision (P) is defined as the proportion of retrieved items judged to be relevant to a query, and the recall (R) is the proportion of relevant items that is retrieved. More formally,

$$P = \frac{\text{Number of relevant items retrieved}}{\text{Total number of retrieved items}}$$

and

$$R = \frac{\text{Number of relevant items retrieved}}{\text{Total number of relevant items in collection}}$$

tic know-how are then unavailable and unconstructible. Simplified linguistic analysis systems, on the other hand, do not produce the expected performance improvements over much simpler weighted keyword systems. It seems that a different approach is needed to carry out retrieval activities in free-text environments. Attempts to provide such an alternative approach are pursued in the remainder of this study.

3 Global Text Matching

Instead of following a weak strategy based on poorly-understood knowledge-base constructions, or on uncertain semantic characterizations obtained from dictionaries and thesauruses, an alternative text handling approach is proposed where the texts themselves form the basis for the analysis operations. The basic idea is that large masses of text are available for analysis, and that the text elements appear within particular contexts that can be used for text interpretation. Assuming that enough text is analyzed, and enough contextual information is used in the analysis, it may be possible to isolate text environments where word meaning and word relationships are reasonably unambiguous. Such homogeneous text neighborhoods could then be linked, and the text links could be and advantageously for text traversal, and retrieval.

The following basic strategy may be considered:

1. Each text is broken down into individual local documents – for example, individual text paragraphs, or text sections.

2. Each local document may be further broken down into local components, such as individual sentences and phrases.

3. Content identifications are assigned globally to the documents and locally to the document components.

4. All documents are compared with all other documents, and when a sufficient coincidence exists between two documents, based on global as well as local text similarity measurements, the two texts are considered to be related.

Such a text matching system is reminiscent of a problem-solving technique known as memory-based reasoning. In memory-based reasoning, one assumes that a global database is available that is loaded with a large number of known prior situations, actions, classifications, desired outputs for particular situations, and so on. Instead of calling on particular inductive or deductive problem-solving techniques when a new problem is encountered, the reasoning system compares the new problem statement with all the known prior cases stored in the database, and the most similar existing case is used to classify the new case.[18] In the same way, a request for information, or an existing text, is compared with all other stored texts, and when sufficient similarities exist between incoming queries and stored texts, one concludes that responses to the queries are at hand.

Given the variability of the language and the ambiguities inherent in natural-language texts, how can one argue that text excerpts with sufficiently similar vocabulary patterns in fact represent semantically homogeneous text environments? For one thing, there are arguments due to some influential philosophers of language. Wittgenstein, in particular, claimed that the meaning of words is best obtained by studying how the words are used in the spoken or written languages:[19]

> "For a large class of cases – though not for all – in which we use the word "meaning", it can be defined thus: the meaning of a word is its use in the language".

By "use", Wittgenstein did not refer exclusively to the word context within which the sentences or utterances are placed, but also to the situations and circumstances in which the utterances appear. But the texts and utterances themselves are surely primary, rather than some external knowledge base or dictionary containing preset definitions. In fact, Wittgenstein specifically warned about the difficulties of providing definitions for many notions in the abstract:[20]

> "Consider, for example, the proceedings we call 'games' – I mean board games, card games, ball games, Olympic games, and so on. What is common to them all? – Don't say there must be something in common or they would not be called 'games' – but look and see .... if you look you will not see anything common to all ...."

Following these arguments, one asks whether by a judicious use of existing texts, the text meaning and text situation can be determined sufficiently well to be able to identify and relate different texts that are semantically similar. Some observers think not: [21]

> "No number of brute linguistic facts (word statistics) can be added up to give us the meaning of a text, where the meaning of a text would include such things as its subject intellectual content, context, use, purpose, or links to other documents."

Such a statement is certainly not controversial. Nevertheless, it is not clear that this collection of "brute linguistic facts" could not lead to a better and more effective text interpretation in an information retrieval setting than any other semantic analysis theory that has heretofore been tried or proposed.

Specifically, regularities will be noticed in actual texts that simplify text comprehension and help elicit appropriate responses from text users. For example, certain semantically distinct texts may very well exhibit superficially similar vocabulary patterns; however it would be most surprising if the local environments of the coinciding vocabulary were also identical since these texts actually cover different subject matter.

Consider as an example the two sample documents (paragraphs) of Fig. 1 extracted from a recent textbook in information science.[22] Apparently, these texts cover related subject matter: the word "text" occurs 5 and 7 times, respectively, in the two paragraphs, and "script" occurs 5 and 9 times, respectively. Eight additional word stems also co-occur in the two sample texts. A global match of the vocabularies in the two paragraphs produces a relatively high global similarity [2] that places the sample paragraph pair at a relatively good rank of 1003 when the 650,000 distinct paragraph pairs of [22] are arranged in decreasing order of their global pairwise similarity. In other words, there are only 1002 other paragraph pairs out of 650,000 with a greater pairwise similarity.

A cursory examination of the sample texts reveals, however, that the global similarity is misleading, because the texts cover quite different subject matter: the term "script" in document 209 refers to a text-editing language by that name, whereas "script" in document 930 represents a particular type of knowledge base used in language understanding. When the local (sentence) contexts are examined, substantial differences become apparent for the two sample texts. The two most similar text sentences, underlined in Fig. 1, are 20902 (sentence 02 in paragraph 209) and 93006 (sentence 06 in paragraph 930). The matching vocabulary and the vocabulary frequencies are "input" (1), "script" (2), and "text" (2), respectively producing a sentence similarity score equal to 5. Such a score is relatively low – the top 2,400 sentence pairs of [22] exhibit frequency scores greater than 7. For the document pair 209-930, all other sentence pairs receive similarity scores lower than 5. Even though some global similarities exist between the words used in the sample paragraphs, the local environment within which the words occur is distinct; one concludes that the texts are probably unrelated.

Analogous considerations are used to conclude that two sample texts are probably related, when reasonably large local similarities are detected, even though the global similarities may not immediately lead to an unequivocal text classification.

The strategy to be followed in the text matching system then consists in comparing text excerpts at various levels of detail: global text similarities applied for example to text paragraphs; more local coincidences within global contexts applied to text sentences; still more local contexts applied to text phrases. When enough similarities are detected in a variety of contexts for particular sample texts, the assumption is that the texts are indeed semantically related.

The methods used for text analysis and for text matching in different text environments are described in the remainder of this study.

[2] The actual text similarity computations are covered later in this study.

4 Automatic Text Indexing

The task of assigning content identifiers to text items is know as indexing. The simplest approach to text indexing consist in using for content representation certain words, or word derivatives, occurring in the individual documents. The following operations may serve for this purpose:[1-3]

1. Take each text item that needs to be separately recognized and retrieved, for example, each text paragraph), and identify the individual text words.
2. Use a dictionary of common function words, also known as a stop list, to delete from the texts the high-frequency words that are insufficiently specific for content representation (and, of, or, but, the, etc.).
3. Use a suffix stripping routine to reduce the remaining words to word stem form; for example, words such as analysis, analyzer, analyzing, etc, would all be reduced to the more general form "analy".
4. For each remaining word stem, compute a term weight reflecting term importance.
5. Represent each text item D_i by the set of word stems, together with the corresponding term weights.

If t terms in all are used to index each document collection, each document D_i is then representable as a vector of terms

$$D_i = (w_{i1}, w_{i2}, ..., ..., w_{it}) \tag{1}$$

where w_{ik} represents the weight of term T_k in document D_i, and a weight of 0 is assumed for terms not assigned to a particular document.

All the steps in the indexing chain are straightforward, except for the term weighting operation. Term weights are used to distinguish terms that are likely to be important for the content representation of particular documents from other terms likely to be less important. This implies that high weights ought to be assigned to words, or word stems, that are prominent in particular documents, but not so prominent elsewhere in the collection.

Two main components of the term weight must be distinguished: the frequency of occurrence of a term T_k in a particular document D_i, also known as the term frequency of T_k in D_i, or tf_{ik}; and a measure that varies inversely with the number of documents in a collection to which a term is assigned, that is, which gives a high value for terms assigned to few documents, and a low value for terms that occur everywhere. The latter has been called the inverse document frequency of term T_k, or idf_k. Typically, idf_k could be computed as $logN/n_k$, where N is the total number of items in a collection, and n_k is the number of items in which T_k occurs.

The term frequency has been used early on as an indicator of term importance in automatic text handling systems.[23] The inverse document frequency was introduced more recently.[24-26] In recent years, the two components have been combined into a single value, known as the $(tf \times idf)$ weight, and computed as $w_{ik} = tf_{ik} \cdot idf_k$ to reflect the importance of T_k in D_i.[28,29] Such a weight provides a high value for terms occurring frequently in an individual document D_i, but rarely on the outside.

It should be noted that the $(tf \times idf)$ weight assigned to a particular term depends crucially on the retrieval environment within which the term assignment occurs. For example, in a textbook retrieval system it may be useful to retrieve individual paragraphs of text. In that case, the term frequency is the frequency of occurrence of a term in an individual text paragraph, and the document or collection frequency is the total number of textbook paragraphs in which the term occurs. A term assigned for paragraph identification can, however also be used to characterize individual text sentences when text sentences are separately retrieved within particular paragraphs. In that case, the weight assignment for the term is very different, because the term frequency is now the occurrence frequency of the term in particular sentences, and the collection frequency is the number of sentences in a paragraph in which the term occurs.

Consider now the computation of a similarity measure between two text items D_i and D_j represented by term vectors of the type shown in expression (1). Assume that the term weight w_{ik} of term T_k in D_i is given either by the term frequency tf_{ik}, or by a combined term frequency and inverse document frequency $(tf_{ik} \times idf_k)$ value. The pairwise text similarity $sim(D_i, D_j)$ may then be computed by using an inner product function

$$Sim(D_i, D_j) = \sum_{k=1}^{t} w_{ik} \cdot w_{jk} \tag{2}$$

or alternatively an inner minimum computation

$$Sim(D_i, D_j) = \sum_{k=1}^{t} min(w_{ik}, w_{jk}) \tag{3}$$

When the text similarity is based on the measurements of expressions (2) or (3), document pairs with many highly weighted matching terms receive large pairwise similarities, and document pairs with few matching terms are assigned low pairwise similarities. When such document pairs are arranged in decreasing order of their pairwise similarity, the pairs at the top of the list (with large pairwise similarities) will all be long documents. Long documents contain many distinct words, some of them with large occurrence frequencies, and such term vectors may produce large pairwise similarities with other documents. On the other hand, short documents with few assigned terms are likely to produce much smaller similarity coefficients.

It is clear that text similarity measurements based on the number and weight of matching terms, that favor long documents at the expense of short ones, cannot be used in situations where nonhomogeneous text items must be treated. In many retrieval environments the document lengths vary considerably; a query-document similarity computation that favors the longer documents over shorter ones could then produce disastrous results. In situations when all documents are equally valuable, normalized term weights must therefore be introduced that reduce all documents to a common length. Typical normalized weights for term T_k in D_i are

$$w_{ik} = \frac{tf_{ik}}{\sqrt{\sum_k (tf_{ik})^2}} \qquad \text{or} \qquad w_{ik} = \frac{(tf_{ik} \cdot idf_k)}{\sqrt{\sum_k (tf_{ik} \cdot idf_k)^2}}$$

When normalized term weights are used, and the similarity between the texts is computed by expressions (2) or (3), the pairwise text similarity ranges between 0 and 1, reflecting the situation when no common terms exist (similarity = 0), and when all terms occur in both documents with equal weight (similarity = 1). When some of the terms coincide but others do not, the pairwise similarity will be greater than 0 but less than 1. Text similarity measurements that include normalization factors are based on the proportion, rather than number, of matching terms. With normalized term weights, large similarities may be assigned to very short documents with few terms, when many of these terms are present also in other short documents.

The text matching experiments described in the next section show that high-quality output is obtainable when the term weights and text similarity measurements are appropriately adjusted to reflect the proper collection environment.

5 Automatic Text Structuring Experiments

When heterogeneous texts are processed that cover a variety of subject matter, it is convenient to impose a structure on the texts such that similarities between related texts are immediately apparent. Texts can be structured in many different ways, depending on the scope of the texts and the particular processing environment. In libraries where the text unit of interest consists of individual books, the collection is usually structured by shelving related items dealing with the same subject matter in adjacent locations within a book stack. Inside the individual books, additional structure might be supplied by relating individual text sections or text paragraph that are known to be semantically related. Text structuring operations may facilitate informal collection browsing as well as more formal retrieval operations.

An obvious approach to text structuring consists in using automatic classification, or clustering, techniques to collect the documents into affinity classes according to perceived similarities in the available descriptions of the texts. Typically, entities that are placed into a common class exhibit greater similarities with other entities inside the same cluster than with entities on the outside. In information retrieval, the use of clustered collections can simplify the collection search operations, because searches are then often confined to a few of the existing clusters, instead of being extended to the whole collection.[29-31]

When stringent clustering criteria are used as in the complete-link clustering strategy, where each clustered item exhibits a high similarity with all other items appearing in the same cluster, the collection is normally divided into a large number of small clusters. On the contrary, when loose clustering criteria apply as in the single-link method, where each item may be closely related to only one other item in the same cluster, a smaller number of much larger clusters is typically generated.

For many data collections, it is difficult to find a clustering strategy that produces just the right number of tightly related classes. Typically, many items will not easily fit into any of the available classes. Such items are normally entered into the closest available class. As a result, some items will appear in common classes even when the subject relationship is not very close.

A finer structured organization is then obtainable by using the similarity measurements introduced in the previous section to compare pairs of items appearing within the same cluster. Text links can then be placed between two clustered texts when sufficient global or local text similarities are detected. The result is a kind of hypertext structure where related texts excerpts are conveniently identified.[32-34]

A simple example of such a structured text organization appears in Fig. 2, where two clusters are shown together with links between certain text pairs. The simple links in Fig. 2 are used to designate document pairs with global text similarities exceeding a stated threshold value. The double links connect document pairs that have significant global text similarities, as well as sufficient local sentence similarities. When structured collection organizations of the type shown in Fig. 2 are available, selective retrieval strategies can be implemented where attention is confined to particular linked document chains within certain clusters only. A search could start with some known relevant document, and follow the strongest document links – for example, from document A in Fig. 2, the search might proceed successively to B, C, D, E, F, G, and H, bypassing documents that are not strongly connected by local as well as global text similarities. Alternatively, structured texts with appropriate links between related text components allow browsing and simplify text traversal by providing diagonal reading paths through the text which can be followed by readers interested in particular topics.

Structured text representations are especially useful for documents such as handbooks, instruction manuals, textbooks, encyclopedias that are inherently structured and are not meant to be read sequentially from one end to the other. In such a case, the individual text components – for example, book paragraphs or dictionary articles – can be linked in accordance with perceived affinities in the corresponding texts. To evaluate the accuracy with which hypertext structures can be automatically generated using sophisticated text matching criteria, experiments were performed using as an experimental database the text of reference [22].

The following questions, in particular, required investigation:

1. how best to compare text paragraphs and text sentences;
2. whether sentence matches are more secure when the sentences appear in matching paragraph pairs;
3. whether semantically related paragraph pairs are more easily identified by individual sentence matches, or by global text matches between the complete texts of the paragraphs;
4. whether matching paragraph pairs are more securely linked when the corresponding paragraphs also contain matching sentence pairs;
5. and whether secure and effective text links can be generated by a vocabulary matching system.

The text of reference [22] is subdivided into 1,140 text paragraphs and about 4,500 text sentences. This implies that there are nearly 650,000 distinct paragraph pairs and over 10 million distinct sentence pairs. In the experiments that were actually performed, all distinct paragraph and sentence pairs are taken into account. However, in a clustered document environment, the comparison operations between distinct items might be restricted to document pairs appearing within identical clusters only. In that case, the number of needed comparisons between paragraph or sentence pairs would of course be drastically reduced.

To evaluate the usefulness of any text linking algorithm, it is necessary to decide whether pairs of texts designated as linked are in fact properly related, that is, whether these texts cover similar subject matter. In the present experiments, samples of linked text pairs (linked paragraphs or linked sentences) were individually examined, and a linking precision measure was computed, reflecting the proportion of properly linked text pairs. When it was reasonable to assume that two linked texts should be retrieved and shown simultaneously to system users, the text link was rated as appropriate; otherwise the link was considered to be extraneous.

For each experiment, the linked text pairs were first ranked in decreasing order of the pairwise similarity, and a sample of 180 text pairs was used to determine the linking accuracy. Of these 180 pairs, 60 pairs were chosen from among those retrieved early in a search, exhibiting the highest pairwise text similarities; 60 more pairs were taken from those ranked above 550 in the ordered list of pairs, and the final 60 were pairs ranked above 1100. This implies that the computed pairwise text similarity is not especially high for 120 out of the 180 examined text pairs in each experiment.

The results for five different text linking experiments are summarized in Table 2. In each case, the correctness of the text linking system is evaluated for two alternative text collections. Each collection includes the (approximately) 2,400 best text pairs, that is, the pairs with the 2,400 highest pairwise text similarities out of the 650,000 available paragraph pairs, and 10 million available sentence pairs defined for the sample collection of reference [22]. As the Table shows, the 180 text pairs actually evaluated are divided into three sets, including 60 pairs that appeared on the list of pairs for the first collection only, 60 more included on the list for the second collection only, and the final 60 appearing on the lists for both collections. This makes possible a separate evaluation of each of the proposed text linking methodologies.

The following principal conclusion are apparent:

1. For the global paragraph comparisons of experiment 1, where different term weighting methods are used before applying the text similarity measure of expression (3), it is clear that the output for collection 21 is far superior to that for collection 24. In the former case, a normalized $(tf \times idf)$ term weight was used, effectively reducing all paragraphs to a common length, whereas a standard term frequency weight tf was used for collection 24 which gives preference to the longer documents.[3] The results show that 53 out of 60 examined paragraph pairs were correctly linked by a global comparison of the normalized texts, but that the linking accuracy slipped to 20 pairs out of 60 without the text normalization. Moreover, no advantage is obtained by using the output produced jointly by both methodologies. It is obvious that length normalization is essential when comparing text paragraphs of varying length and scope.

2. When text sentences are compared instead of paragraphs, the results are reversed; that is, for sentence-length excerpts, the text similarity is best computed by a using an unnormalized term-frequency similarity measure. The problem with normalized similarity measurements for sentence comparisons is the large number of small sentence fragments included in ordinary running text. When normalized similarity measures are used, based

[3] In the experiments, the normalized $(tf \times idf)$ weight is computed by using an enhanced term frequency measure set to produce term frequency values between 0 and 1, defined as $(0.5 + 0.5 \frac{tf_{ik}}{max\, tf_{ik}})$, where tf_{ik} is the number of occurrences of term T_k in D_i.[34] The total weight of term T_k in document D_i is then obtained as

$$w_{ik} = \frac{(0.5 + 0.5\frac{tf_{ik}}{max\, tf_{ik}})(log\,\frac{N}{n_i})}{\sqrt{\sum_{vector\, D_i}(0.5 + 0.5\,\frac{tf_{ik}}{max\, tf_{ik}})^2(log\,\frac{N}{n_i})^2}}. \quad (4)$$

on the proportion of matching terms, it is easy to find dozens of sentences, or sentence fragments, characterized by the same two or three significant terms. In such cases, the proportion of matching terms may be 100 percent, and the similarity between them is equal to 1, even though such sentences may have little real meaning.

For collection 15, the top 303 sentence pairs out of 2,400 exhibit perfect pairwise similarities of 1, and many additional nonsignificant sentence pairs are assigned high similarities somewhat smaller than 1 when normalized text similarity measurements are used. Examples are:

"For the example of Fig. 1" and "See for example Fig. 1" (similarity 1.0).

"There are three possibilities" and "Several possibilities present themselves" (similarity 0.7197).

"Alternatively one can write" and "11.3 (a) would be written as" (similarity 0.7167).

Overall only about 17 percent of the sentence links are acceptable for collection 19, compared with 57 percent of the links for collection 3 which is constructed using simple term frequency weights where the number of term matches is controlling. By far the best sentence matching results are obtained by using a combination of techniques requiring both a high global (normalized) similarity, as well as a reasonable number of individual term matches. The right-most column of Table 2, Section 2, shows that 85 percent of the sentence links are acceptable for sentence pairs that are linked by both the strategies used for collections 3 and 19.

3. In the preceding experiment, the sentence pairs were freely chosen from any of the existing document pairs. In sentence matching environments, it is important also to compare the overall accuracy of a global comparison of sentence pairs with the accuracy of sentence comparisons for sentences known to occur in already matching paragraph pairs. The corresponding evaluation data are shown in Table 3, Section 3, where collection 17 covers the global sentence pair match using the top 2,400 sentence pairs regardless of sentence origin, and collection 31 includes only sentences from document pairs in collection 21. The output of Table 2 shows that the overall accuracy of sentence matching reaches only about 30 percent for the global comparisons of collection 17, whereas a reasonably high accuracy of 82 percent is obtained for collection 31. Once again the joint use of the global sentence comparisons as well as comparisons within already matching documents (joint output for collections 17 and 31) does not produce improvements over the use of collection 31 alone.

4. Sections 2 and 3 of Table 2 cover text linking and retrieval experiments, where text sentences are independently manipulated. In practice, it is unlikely that short excerpts such as sentences would be retrieved by themselves without additional context. A sentence linking system could then be treated as an enhancement of paragraph retrieval that may improve the accuracy of paragraph linking rather than as an independent activity. In the experiment of Section 4 of Table 2, an attempt is made to decide whether paragraph comparisons are more secure when the corresponding paragraphs also contain related lower-level constructs such as matching sentences.

   The results of Table 2 indicate that the output for paragraphs also containing matching sentences (21 and 31sd) is far superior to the output obtained with the global paragraph matches alone (21 only). The middle column in Section 4 of Table 2 is empty, because all document pairs specified in collection 31sd are known in advance to occur also in collection 21.

5. A final analysis for paragraph pairs containing matching text sentences appears in Table 2, Section 5, covering comparisons between the global paragraph pairs of collection 24, and the paragraph pairs containing matching sentences of collection 31sd. In this case, different text indexing and retrieval methods are compared: collection 31sd is a subset of collection 21 based on ($tf \times idf$) term assignments, whereas collection 24 uses tf weights alone. The right-hand column of Table 2, Section 5 shows that the text linking results are practically perfect when the strengths of different systems are superimposed. The output in the right-hand column of Table 2, Section 5 represents paragraph pairs that have both a high global ($tf \times idf$) normalized text similarity (collection 31sd), as well as the large number of matching term pairs required by the similarity measure used with collection 24. These document pairs also contain at least one pair of matching text sentences.

In summary, the text linking experiments using textbook paragraphs and sentences produce a high-order of linking accuracy for global paragraph matching based on normalized ($tf \times idf$) similarities (see Section 1 of Table 2). When global free-text sentences are compared, reasonable accuracies (82 percent) are obtainable by insisting both on a minimum number of matching terms (unnormalized tf similarity), as well as a high proportion of matching terms (normalized $tf \times idf$ similarity). When both criteria are satisfied, all perfectly matching short sentence fragments are rejected. Alternatively, secure sentence comparisons are also obtainable by confining the matching operations to sentences occurring in previously identified matching paragraphs.

When complete text paragraphs are used as linking and retrieval units, the text linking is more secure when matching text sentences are also included in these paragraph (88 percent versus 75 percent), and near-perfect results are obtained when several requirements are met. In particular, a linking accuracy of 98 percent was obtained for paragraph pairs including at least one pair of matching sentences, characterized by both a large proportion of matching terms, as well as a sufficient number of individual term matches.

The insights gained through the experiments of Table 2 are used in the next section for the design of retrieval system for electronic-mail messages.

6 The Retrieval of Electronic Mail Messages

The automatic processing of electronic mail (e-mail) is becoming increasingly important to a large community of users. Hundreds of thousands of e-mail messages are currently circulating through communications networks, covering a wide variety of topics in many different styles and completely unrestricted vocabulary. Unfortunately, a usable message retrieval system has not so far been implemented in operational situations. For all practical purposes, it is necessary to scan message collections sequentially in order to find particular items.

As the examples of Fig. 3 show, messages are subdivided into four main parts:

1. A header portion giving sender and receiver addresses, dates, an assigned message identifier for the current message, a newsgroup identifier, a subject identifier, and additional routing information.

2. An optional quotation from one or more previous messages to which the current message may be related. In the illustration of Fig. 3, the message of Fig. 3(b) contains a quotation from the earlier message of Fig. 3(a).

3. The body of the message consisting of unrestricted text.

4. A message signature often supplemented by additional comments.

In the message retrieval experiment described in the present study, the text portions quoted from other messages have been disregarded (except where otherwise noted). Obviously these quotations will drastically affect any text matching system, and they cannot be used in a fair assessment of retrieval effectiveness. Message signatures have also been disregarded wherever possible; often these signatures are loaded with additional wisdom, and extraneous statements that have nothing to do with the message content. Finally, the header information is also disregarded for text matching purposes, so that message retrieval depends on a comparison of the text bodies alone.

Experiments conducted with electronic-mail messages are appropriate not only because of the obvious interest in message retrieval, but also because objective relevance assessments are available which make it possible to compute evaluation parameters without resorting to the use of subjective relevance assessments obtained from the outside. For present purposes, the assumption is that a stored message is relevant to a particular query message when either the "Subject" lines in the respective headers are identical, or when the "Reference" line in the header of a message contains a reference to the query message.

Both of these conditions are satisfied in the example of Fig. 3: the subject lines in the headers are identical, and the reference line in document 318 contains the identifier for the query message 18. Hence document 318 is assumed to be relevant to query 18.

While the use of objective relevance assessments is a great convenience in system evaluation that may help in producing reliable evaluation data, the objectively determined relevance assessments of documents with respect to queries are not always trustworthy. In the present case, the available objective relevance assessments for e-mail messages are too strict: for example, a relevant document may be called nonrelevant with respect to a query because of slight variations in the respective subject descriptions. When such an apparently nonrelevant document is retrieved early in a search (for example document 1076 "PA2 key" with respect to query 71 "PA2 disable"), the result is scored as the false retrieval of a nonrelevant item wrongly depressing the retrieved precision.

Another relevance problem arises when chains of messages exist that all carry the same subject identification, and hence are all objectively relevant to a given query. In such a case, a document that actually replies to the query is in fact relevant. But subsequent documents may refer not to the query at all, but to replies to the query, or to replies to replies to the query, all the while maintaining the same subject identification. Typically, the subject matter shifts, and these later documents are not in fact relevant to the original query, although the system says that they are. When these later documents are not retrieved, or are retrieved late in a search, the recall is depressed.

Still, the immediate availability of relevance assessments of all document with respect to all queries is obviously attractive, and much preferable to a manual generation of thousands of individual human judgments. This is the reason for utilizing such objective relevance standards in the present study.

A collection of 1984 messages is used for experimental purposes together with 180 search queries. The queries are simply special messages sent out to request information from network users. The retrieval system compares each document to each query, using the previously described global or local text matching systems, and a full ranking is obtained of all the documents with respect to each query in decreasing order of the query-document similarity. Because the retrieval rank of each relevant document with respect to each query is known, it is easy to generate the needed recall-precision data.[1-3]

A recall-precision table appears in Table 3(a) giving the average precision values for 180 queries at certain fixed values of the recall for the message collections used with and without the quoted text portions. In the output of Table 3, only global text matches are used, based on the normalized ($tf \times idf$) term weights of expression (4), and no local sentence matching is involved. The output obtained for the quoted collection in Table 3(a) is nearly perfect: when the recall is 0.5 (that is, after half the relevant items are retrieved for each query), the precision still remains above 0.9 (that is, the proportion of nonrelevant retrieved material is less than 10 percent).

A variable retrieval cut-off is used for the various queries to generate the recall-precision data of Table 3(a). Specifically, recall and precision values are computed for each query after each relevant message is retrieved, and the computation stops when the last relevant item has been retrieved and the recall has reached 1.0 for each query. This means that the number of retrieved documents used in the recall-precision calculations varies greatly from query to query.

A different retrieval strategy is used to generate the performance figures of Table 3(b). Here exactly 10 (or exactly 30) items are retrieved for each query, regardless of the number of relevant items included in the collection for that query. The same outstanding performance already noted in Table 3(a) for the quoted collection is again present in the recall data of Table 3(b). When 10 items are retrieved for each query, the recall is almost 90 percent for the quoted collection, indicating that nearly all relevant items in the collection have been retrieved. The fixed cut-off evaluation of Table 3(b) necessarily produces much lower precision that the variable cut-off process used in Table 3(a). For 117 of the 180 test queries only one or two relevant documents are included in the collection. Many of these relevant items are retrieved at the head of the ranked output list; nevertheless at least 10 items are considered for each query in the output of Table 3(b) most of which will necessarily be nonrelevant. This effectively limits the attainable precision performance for each query to 0.2 for queries with 2 relevant items, and 0.1 for queries with one relevant item.

Not much needs to be done to improve the performance for the quoted collection. The retrieval performance is not as outstanding for the message collection used without quotations. Even there, very respectable figures are obtained. When the recall is 0.50, the precision is 0.58 (see Table 3(a)) indicating that only about 40 percent of the retrieved items are nonrelevant on average when 50 percent of the relevant have been retrieved for each query. The fixed cut-off system of Table 3(b) indicates moreover that the recall exceeds 68 percent even for the collection without quotations at a fixed retrieval level of 10 items. For the unquoted collection, there is, however, room for improvement. Accordingly, the local sentence-matching process is invoked in that case, and evaluation data are presented for message retrieval based on global text matching supplemented by local sentence-matching methods. The corresponding results appear in Table 4.

The performance data once again reflect the average performance for 180 queries. A semi-fixed retrieval threshold is used to obtain the data of Table 4, consisting either of 20 retrieved items for each query, or the retrieval rank of the last relevant item, whichever comes earlier. Thus when the rank of the last relevant item is 5, only 5 documents are considered in the recall-precision computation; on the other hand, 20 items are examined for each query, when the rank of the last relevant items is 20 or higher. The first line of Table 4 shows that for the 180 queries a total of 2071 items are retrieved using only the global text match without local sentence comparison; 370 of these items are relevant, producing a retrieval precision of 0.1787.

Sections b) and c) of Table 4 cover the experiments where at least one, or at least two, matching sentence pairs are used for retrieval in addition to the sufficiently high global text match. Normalized ($tf \times idf$) term weights (as in expression (4)) are used for the global text matching operation. For the sentence matches an inner minimum similarity is used based on term-frequency (tf) weights, as shown in expression (3). Evaluation data are presented for various values of the sentence similarity measure, ranging from 2 to 6 in each case. When an appropriate number of matching sentences is detected, a modified similarity measure is computed that includes both global as well as local sentence matching criteria:

$$New\, sim = Old\, sim[1 + Number\, of\, Matching\, Sentences]. \qquad (5)$$

Here "old sim" is the global text matching similarity, and "new sim" the combined similarity measure for message pairs with matching sentences.

The figures in Table 4 show that the sentence-match requirement operates as a precision device. Fewer items are retrieved, but the proportion of relevant items among the retrieved ones is much larger when sentence matches are required, especially so when the sentence matching coefficients are large. The retrieval precision for the semi-fixed retrieval threshold used in the evaluation increases from 0.1787 without local sentence match to 0.5426 when at least one matching sentence pair is found with sentence similarity values of at least 6, and to 0.6667 for two such matching sentence pairs. It is clear that the local text matching system substantially enhances the power of free-text retrieval systems based on text matching methodologies. Given the variability of the input texts, and the strictness of the available relevance assessments, the retrieval effectiveness obtainable by using text matching environments of different scope is quite substantial.

In this study, only two text matching contexts have been taken into account, namely paragraphs and sentences. Additional refinements might be added – for example by matching individual phrases within matching sentences, or restricting the collection search to items located within the same text clusters, instead of extending the search globally to the whole collection. Improved term and phrase weighting methods might also produce somewhat enhanced performance measurements.

The text matching system with variable contexts introduced in this study presents the following advantages:

- it is robust and applicable to large collections in any subject area;

- it is immediately usable, and does not depend on the availability of dictionaries and thesauruses, or on the semantic characterization of text components;
- it provides a high order of performance, and this precision performance may be improvable by using additional matching criteria for lower-level constructs;
- no other text-analysis and retrieval system is currently in view that competes with the effectiveness of the proposed methodology for general purpose texts in arbitrary subject areas.

7 References

1. G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval*, McGraw Hill Book Co., New York, NY, 1983.
2. F.W. Lancaster, *Information Retrieval Systems – Characteristics, Testing and Evaluation*, Second Edition, J. Wiley and Sons, New York, 1979.
3. C.J. van Rijsbergen, *Information Retrieval*, Second Edition, Butterworths, London, 1979.
4. D.C. Blair and M.E. Maron, An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System, *Communications of the ACM*, 28:3, March 1985, 289-299.
5. R.M. Tong, L.A. Appelbaum, V.N. Askman, and J.F. Cunningham, Conceptual Information Retrieval Using RUBRIC, *Proceedings of the Tenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, C.T. Yu and C.J. van Rijsbergen, editors, Association for Computing Machinery, New York, June 1987, 247-253.
6. P.S. Jacobs and L.F. Rau, Natural Language Techniques for Intelligent Information Retrieval, *Proceedings of the Eleventh Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Y. Chiaramella, editor, Association for Computing Machinery, New York, June 1988, 85-99.
7. D. DeJaco and G. Garbolino, An Information Retrieval System Based in Artificial Intelligence Techniques, *Proceedings of the Ninth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, F. Rabitti, editor, Association for Computing Machinery, New York, June 1986, 214-220.

8. A.F. Smeaton and C.J. van Rijsbergen, Experiments in Incorporating Syntactic Processing of User Queries with a Document Retrieval Strategy, *Proceedings of the Eleventh Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, Y. Chiaramella, editor, Association for Computing Machinery, New York, June 1988, 31-51.

9. J. Fagan, The Effectiveness of a Nonsyntactic Approach to Automatic Phrase Indexing for Document Retrieval, *Journal of the American Society for Information Science (ASIS)*, 40:2, March 1989, 115-132.

10. M. Dillon and A.S. Gray, Fully Automatic Syntactically Based Indexing Systems, *Journal of the American Society for Information Science (ASIS)*, 34:2, March 1983, 99-108.

11. K. Sparck Jones, *Automatic Keyword Classification for Information Retrieval*, Butterworths, London, 1971.

12. P. Schauble, Thesauruses-Based Concept Spaces, *Proceedings of the Tenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, C.T. Yu and C.J. van Rijsbergen, editors, Association for Computing Machinery, New York, June 1987, 254-262.

13. D.E. Walker and R.A. Amsler, The Use of Machine-Readable Dictionaries in Sublanguage Analysis, in *Analysing Language in Restricted Domains*, R. Grishman and R. Kittredge, editors, L. Erlbaum, Hillsdale NY, 1986.

14. E.A. Fox, J.T. Nutter, T. Ahlswede, M. Evens and J. Markowitz, Building a Large Thesaurus for Information Retrieval, *Proceedings Second Conference on Applied Natural Language Processing*, Association for Computational Linguistics, Austin TX, February 1988, 101-108.

15. G. Salton, Another Look at Automatic Text-Retrieval Systems, *Communications of the ACM*, 29:7, July 1986, 648-656.

16. J.L. Fagan, Experiments in Automatic Phrase Indexing: A Comparison of Syntactic and Nonsyntactic Methods, Doctoral Dissertation, Cornell University, Technical Report 87-868, Department of Computer Science, Cornell University, Ithaca NY, September 1987.

17. J.L. Fagan, Automatic Phrase Indexing for Document Retrieval: An Examination of Syntactic and Nonsyntactic Methods, *Proceedings of the Tenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, C.T. Yu and C.J. van Rijsbergen, editors, Association for Computing Machinery, New York, June 1987, 91-101.

18. C. Stanfill and D.L. Waltz, Toward Memory-Based Reasoning, *Communications of the ACM*, 29:12, December 1986, 1213-1228.

19. L. Wittgenstein, *Philosophical Investigations*, Basil Blackwell and Mott, Ltd., Oxford, 1953, paragraph 43.

20. L. Wittgenstein, *op. cit.*, paragraph 66.

21. D.C. Blair, *Language and Representation in Information Retrieval*, Elsevier Science Publishers, New York, 1990.

22. G. Salton, *Automatic Text Processing - The Transformation, Analysis, and Retrieval of Information by Computer*, Addison Wesley Publishing Company, Reading MA, 1989.

23. H. P. Luhn, A Statistical Approach in the Mechanized Encoding and Searching of Literary Information, *IBM Journal of Research and Development*, 1:4, October 1957, 309-317.

24. K. Sparck Jones, A Statistical Interpretation of Term Specification and its Application in Retrieval, *Journal of Documentation*, 28:1, March 1972, 11-21.

25. S.E. Robertson and K. Sparck Jones, Relevance Weighting of Search Terms, *Journal of the American Society for Information Science (ASIS)*, 27:3, 1976, 129-146.

26. W.B. Croft and D.J. Harper, Using Probabilistic Models of Information Retrieval Without Relevance Information, *Journal of Documentation*, 35:4, December 1975, 285-295.

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

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

29. A. Griffiths, L.A. Robinson, and P. Willett, Hierarchic Agglomerative Clustering Methods for Automatic Document Classification, *Journal of Documentation*, 40:3, September 1984, 175-205.

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

31. W.B. Croft, Clustering Large Files of Documents Using the Single Link Method, *Journal of the American Society for Information Science (ASIS)*, 28:6, November 1977, 341-344.

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

33. D.R. Raymond and F.W. Tompa, Hypertext and the New Oxford Dictionary, *Communications of the ACM*, 31:7, July 1988, 871-879.

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

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

Type of Content Representation	CACM Collection 3204 documents 52 queries	CISI Collection 1460 documents 76 queries
Weighted single term indexing [15]	0.2604	0.2450
Single terms plus phrases obtained by statistical co-occurrence	0.3195 (+22.7%)	0.2503 (+2.2%)
Single terms plus phrases obtained by syntactic analysis output	0.2830 (+8.7%)	0.2480 (+1.2%)

Table 1: Comparison of Retrieval Performance Obtained by Using Single Terms and Phrase Indexing (from [16])

	Type of Experiment	Proportion of Acceptable Links between Text Pairs		
1.	Paragraph Comparison Methods			
	Collection 21: 2400 paragraph pairs; normalized ($tf \times idf$) paragraph similarily	21 only 53/60 (88%)	24 only 20/60 (33%)	21 and 24 53/60 (88%)
	Collection 24: 2400 paragraph pairs; (tf) similarity not normalized for length			
2.	Sentence Comparison Methods			
	Collection 3: 2394 sentence pairs with (tf) similarity ≥ 7	3 only 34/60 (57%)	19 only 10/60 (17%)	3 and 19 51/60 (85%)
	Collection 19: 2400 sentence pairs; normalized ($tf \times idf$) sentence similarity			
3.	Sentence Comparison within Matching Document			
	Collection 17: 2400 sentence pairs with phrase indexing; (tf) sentence similarity	17 only 18/60 (30%)	31 only 53/60 (88%)	17 and 31 49/60 (82%)
	Collection 31: 2177 sentence pairs within top 2400 document pairs of collection 21 with phrase indexing; (tf) sentence similarity			
4.	Paragraph Linking with and Without Matching Sentences			
	Collection 21: 2400 paragraph pairs; normalized ($tf \times idf$)	21 only 45/60 (75%)	31sd only –	21 and 31sd 53/60 (88%)
	Collection 31sd: 2177 paragraph pairs included in collection 21 containing at least one matching sentence pair from collection 31			
5.	Mixed Method Paragraph Linking			
	Collection 24: 2400 paragraph pairs; tf similarity not normalized	24 only 32/60 (53%)	31sd only 54/60 (90%)	24 and 31sd 59/60 (98%)
	Collection 31sd: 2177 paragraph pairs from collection 21 containing a matching sentence pair from collection 31			

Table 2: Evaluation of Paragraph and Sentence Pair Linking for Textbook Excerpts

Recall	Average Precision for 180 Queries Message Collection With Quoted Passages	Message Collection without Quotations
0.1	0.9480	0.6724
0.3	0.9279	0.6402
0.5	0.9016	0.5806
0.7	0.8097	0.4348
0.9	0.7188	0.3644

a) Recall-Precision Table for E-Mail Collection
(1984 messages, 180 queries)

	Performance After 10 Retrieved Items		Performance After 30 Retrieved Items	
	Recall	Precision	Recall	Precision
Message Collection with Quotations	0.8912	0.2378	0.9291	0.0869
Message Collection without Quotations	0.6836	0.1767	0.8033	0.0731

b) Evaluation with Fixed Retrieval Cuttoff for E-Mail Collection
(1984 messages, 180 queries)

Table 3: Retrieval Evaluation for Electronic-Mail Collection
(global text matching only; no sentence matching)

		Total Number of Retrieved Items	Total Number of Retrieved Relevant Items	Precision
a)	no sentence matching	2071	370	0.1787
b)	one matching sentence-pair required			
	sentence similarity 2	1420	333	0.2345
	sentence similarity 3	599	217	0.3623
	sentence similarity 4	280	128	0.4571
	sentence similarity 5	150	75	0.5000
	sentence similarity 6	94	51	0.5426
c)	two matching sentence-pairs required			
	sentence similarity 2	888	281	0.3164
	sentence similarity 3	296	135	0.4561
	sentence similarity 4	109	66	0.6055
	sentence similarity 5	49	31	0.6327
	sentence similarity 6	27	18	0.6667

Table 4: Retrieval Evaluation for Electronic-Mail Collection
with Sentence-Pair Comparisons
(1984 messages, 180 queries)

I.209
4.4 Typical Processing Systems
4.4.1 Off-Line Text Editing System
The SCRIPT system, a typical editing and formatting facility, uses a text editor to create input files consisting of text elements interspersed with SCRIPT commands.[7] In SCRIPT, detailed low-level formatting instructions can be supplied, and formatting specifications are directly specified by the user. Table 4.1 gives a list of typical SCRIPT commands; a few simple formatting examples are shown in Table 4.2. The basic commands appear in a .xy form, and commands may be used singly, or in paired sets. In the latter case, the two appearances are used to bracket the text portion affected by the command. As shown in Tables 4.1 and 4.2, SCRIPT includes all the usual formatting instructions, such as margin specification, indentation, underscoring, and spacing. In addition, complete text blocks can be identified and processed as units.

a) Paragraph 209 of Chapter 4

.I 930
A last approach to text summarization consists of using stored scripts of important facts characterizing particular situations.[33—35] Each script is a repository of a restricted set of events that may occur, or that are related to particular contexts. Thus an automobile-accident script may contain information about the type of vehicle involved, the objects involved in a collision with a vehicle, the location of the accident, the type of injuries sustained, the number of people involved, the fault determination, if any, and so on. Because each script is specific and circumscribed, the text analysis to relate text portions to particular scripts may be relatively shallow. In particular, dictionaries may specify context tests that must be satisfied by text elements if the dictionary entry is to be applied to particular text excerpts. After applicable scripts are identified, the script structures can be used to produce formalized paraphrases of the story. When available input matches script information closely enough – in the sense that a sufficient number of relevant words and phrases are found in the text – a correct choice of useful scripts may then lead to construction of reasonable text summaries. This solution is once again likely to produce questionable output for texts that straddle different scripts, or texts that lie somewhat outside available knowledge structures.

b) Paragraph 930 of Chapter 12

Figure 1: Two Sample Textbook Paragraphs (from [22])

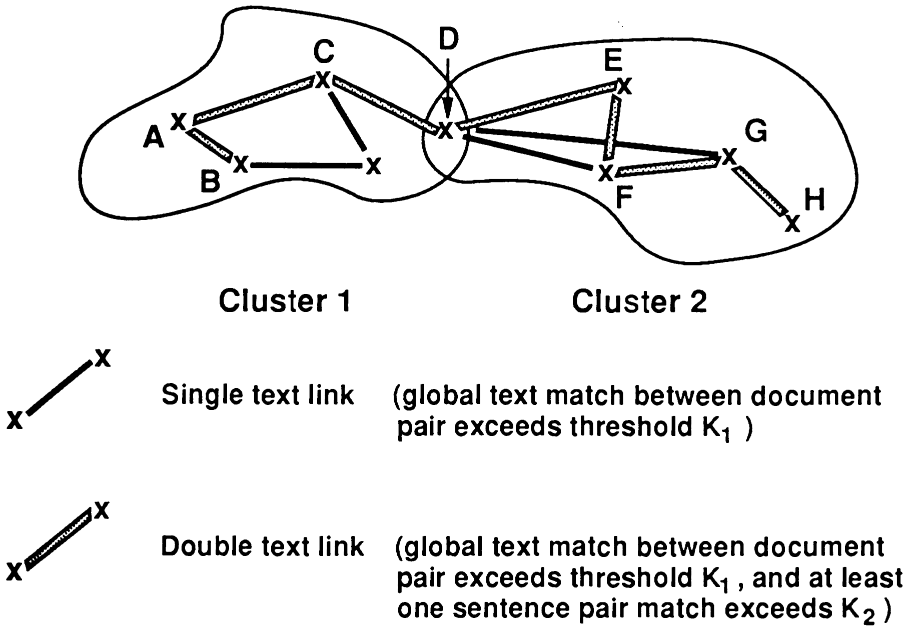

Figure 2: Structured Text Organization

query 18
Document path: text/lang/c/23649 18
Relay-Version: version B 2.10.3 4.3bsd-beta 6/6/85; site cornell.UUCP
Path: cornell!mailrus!shadooby!ginosko!gem.mps.ohio-state.edu!apple!motcsd!
From: swh@hpcupt1.HP.COM
Newsgroups: comp.lang.c
Subject: <varargs.h> for Turbo C ?
Message-ID: <5940011@hpcupt1.HP.COM>
Date: 12 Oct 89 16:01:26 GMT
Date-Received: 13 Oct 89 10:06:16 GMT
Organization: Hewlett Packard, Cupertino
Lines: 17
<varargs.h> for Turbo C ?
The Turbo C compiler product does NOT supply <varargs.h> in its include libraries. (Perhaps because it is not ANSI enough?)
Has anyone developed a suitable <varags.h> file for use with Turbo C?
And, if so, would you e-mail me a copy, or better yet, post it.
This would be very helpful when porting C code from a U**X environemt.

swh@hpda.hp.com
... hplabs!hpda!swh
HPG200/11
(408) 447-5580

a) Typical Query Message (Query 18)

Doc 318
Document path: text/lang/c/23645 318
Relay-Version: version B 2.10.3 4.3bsd-beta 6/6/85; site cornell.UUCP
Path: cornell!mailrus!tut.cis.ohio-state.edu!ucbvax!agate!saturn!sidney
From: sidney@saturn.ucsc.edu
Newsgroups: comp.lang.c
Subject: Re: <varargs.h> for Turbo C ?
Message-ID: <9406@saturn.ucsc.edu>
Date: 13 Oct 89 07:11:26 GMT
Date-Received: 13 Oct 89 09:49:18 GMT
References: <5940011@hpcupt1.HP.COM>
Reply-To: sidney@saturn.ucsc.edu
Organization: University of California, Santa Cruz
Lines: 10
In article <5940011@hpcupt1.HP.COM> swh@hpcupt1.HP.COM writes:
><varargs.h> for Turbo C ?
>The Turbo C compiler product does NOT supply <varargs.h> in its include.
My Turbo C 2.0 reference manual says that the va_... stuff is in stdarg.h, so you don't need any varargs.h. A quick glance at stdarg.h showed definitions for va_start, va_arg, va_end, and the type va_list.
– <sidney@saturn.ucsc.edu>

b) Answer to Query 18 (Document 318)

Figure 3: Typical Query-Document Pair in E-Mail Collection