ELECTRONIC PUBLISHING, VOL. 2(4), 179–192 (DECEMBER 1989)

Paragraph-based nearest neighbour searching in full-text documents

SULIMAN AL-HAWAMDEH AND PETER WILLETT

Department of Information Studies,
University of Sheffield,
Western Bank,
Sheffield, S10 2TN, UK

SUMMARY

This paper discusses the searching of full-text documents to identify paragraphs that are relevant to a user request. Given a natural language query statement, a nearest neighbour search involves ranking the paragraphs comprising a full-text document in order of descending similarity with the query, where the similarity for each paragraph is determined by the number of keyword stems that it has in common with the query. This approach is compared with the more conventional Boolean search which requires the user to specify the logical relationships between the query terms. Comparative searches using 130 queries and 20 full-text documents demonstrate the general effectiveness of the nearest neighbour model for paragraph-based searching. It is shown that the output from a nearest neighbour search can be used to guide a reader to the most appropriate segment of an online full-text document.

KEY WORDS Best match searching Browsing Full-text documents Information retrieval Nearest neighbour searching Paragraph-based searching

FULL-TEXT DOCUMENT RETRIEVAL

The last few years have seen a rapid growth in the availability and use of computer software packages for the storage and retrieval of documents from textual databases[1,2]. These packages have usually, but not exclusively, been designed for the processing of database records that contain fairly short sections of text. The two main types of database are reference retrieval systems, in which each record contains a bibliographic citation to a journal article, technical report, conference paper, etc, and library catalogues describing the monograph and periodical holdings of a library system, in which each record contains author, publisher and title data, together with classification codes, subject headings, etc. which have been added by indexers or cataloguers[3,4].

The widespread use of word processing and electronic publishing systems has led to a huge increase in the number of full-text databases available in-house and from public database hosts. Examples include the full-texts of newspapers, laws and judicial proceedings, reference books, e.g., encyclopaedias and software documentation, and minutes of meetings; thus retrieval software must increasingly provide facilities for the storage and searching of full-text documents. However, the same basic retrieval mechanisms are used for full-text searching as for shorter textual records such as those discussed above (which we shall refer to subsequently as *summary records*). This is despite the fact that there have been many studies which suggest that the characteristics of full-text databases are very different from those of the more traditional types of

0894–3982/90/040179–14$?.00

Received 24 June 1989
Revised 30 January 1990

textual database[5–12]. A search requirement specific to full-text databases is the ability to browse around *within* a single document to identify just those (small) portions that are relevant to the user's request, a facility which O'Connor refers to as *passage retrieval*[13,14]. Such a facility is necessary owing to the fact that a full-text document may well be thousands, or tens of thousands, of lines long and thus not amenable to rapid scanning at a terminal as with summary records [11,15–17].

A full-text document is made up of a number of paragraphs, each of which is of the same order of size as the summary records in a conventional textual database. It would thus seem possible that while full-text documents are very different from summary records, the individual paragraphs comprising a single full-text record are not so very different; accordingly, it should be possible to use retrieval techniques of proven utility in searching databases of summary records for searching the paragraphs comprising a full-text document. The utility of paragraph-based access to full-text has been noted previously by Cohen *et al*.[18] who were involved with the design of an experimental system at Chemical Abstracts Service to assist indexing and abstracting staff in the task of subject analysis. Paragraphs were found to be the the most useful level of description for providing interactive access to sections of full-text documents. More recently, Tenopir[19] has emphasized the general effectiveness of paragraph-level searching in a comparison of different retrieval strategies for full-text databases.

In this paper, we evaluate the use of the *nearest neighbour* searching model[1,3], for paragraph-based access to full-text documents. We shall first describe this model and compare it with the better established Boolean retrieval model. We then outline the experimental system we have used together with the details of the queries and full-text documents that have formed the basis for our comparison of Boolean and nearest neighbour searching; the results of this extended comparison are discussed in a separate section. The next section then proposes an alternative type of nearest neighbour search that can guide a user to that part of a full-text that is likely to contain the greatest amount of relevant material. The paper concludes with a summary of our findings and suggestions for future work.

NEAREST NEIGHBOUR SEARCHING IN FULL-TEXT DOCUMENTS

The overwhelming majority of current text retrieval systems are based on the well known Boolean retrieval mechanism in which a query contains a set of keywords linked together with the Boolean logical operators AND, OR and NOT (in addition, truncation and proximity operators are available to refine a search)[1,3]. This model would thus seem to provide an obvious starting point for the provision of browsing facilities within a full-text document. However, despite its popularity, there are several problems associated with the Boolean retrieval mechanism, especially if end-users are to be able to carry out searches of text databases for themselves [20–22].

- Without a fair degree of training, it is difficult to formulate any but the simplest of queries using the Boolean operators AND, OR and NOT. Accordingly, trained intermediaries often carry out a search on behalf of the user with the actual information need.
- There is very little control over the size of the output produced by a particular

query. Without a detailed knowledge of the contents of the file, the searcher will be unable to predict *a priori* how many records will satisfy the logical constraints of a given query. Accordingly, several formulations of a query may be needed before an acceptable volume of output is produced.

- Boolean retrieval provides no mechanism by which the documents can be ranked in order of decreasing probability of relevance: instead, documents are either retrieved or not retrieved.
- There are no obvious means by which one can reflect the relative importance of different components of the query, since Boolean searching implicitly assumes that all of the terms in the system have weights of either unity or zero, depending upon whether they happen to be present or absent in the query.

These limitations have been noted in searches of databases of summary records, but similar comments also apply to searches within individual full-text documents.

Many of the problems can be overcome by use of the *nearest neighbour*, or *best match* searching technique [3,21–23]. A nearest neighbour search matches a set of query words against the sets of words corresponding to each of the documents in the database, calculates a measure of similarity between the query and each document, and then sorts the documents into order of decreasing similarity with the query. The output from the search is a ranked list, in which those documents that the system judges to be most similar to the query are displayed first to the user. Accordingly, if a sensible measure of similarity has been used, the first documents inspected will be those which have the greatest probability of being relevant to the query which has been submitted. Retrieval systems based on nearest neighbour searching can help to alleviate many of the problems associated with Boolean searching.

- There is no need to specify Boolean interconnections between the keys in the query, since a nearest neighbour search requires just an unstructured list of keywords.
- The ranking of the database in response to the query allows complete control over the amount of output which needs to be inspected, since the user can search down the list just as far as is needed.
- It is normally very easy to take weighting information into account when calculating the degree of similarity between the query and the documents in the file. Moreover, these weights may derive from user judgements of relevance for previously inspected output, and there is hence an attractive mechanism available for the automatic incorporation of relevance feedback information if a second search is required.

The main limitation of nearest neighbour searching is that the absence of the logical operators precludes the explicit specification of phrasal (AND) and synonymic (OR) relationships between query terms.

Nearest neighbour retrieval systems for summary records are becoming widely used[21,22]. To date, however, there has been rather less interest in the use of this retrieval mechanism for searching full-text documents. Ro[7,8] has considered the ranking of *entire* full-text documents as a post-processing stage after a Boolean search has been carried out; thus, the user inputs a conventional Boolean query and the documents retrieved from this search formulation are then displayed in descending order of similarity with the query. There is, however, no attempt to order the components of individual documents within this output. The only attempt, of which we are aware, to carry out such a ranking is that described by Bernstein and Williamson[16] in the context of the ANNOD system. These workers discuss the application of a nearest neighbour search strategy to the ranking of sections of text in the Hepatitis Knowledge Base. This is a critically evaluated body of writings on the subject of viral hepatitis that has been compiled as a collaborative effort by experts in hepatitis research and that has then been organized into a hierarchical structure for retrieval purposes. Specifically, the information in ANNOD is arranged under a series of topic headings and subheadings, each of which has a ‘synthesis’ paragraph summarizing the information that is contained in the following subordinate paragraphs. Bernstein and Williamson’s results are impressive but seem to depend, in part, on the extensive manual pre-processing of the text that is carried out and on the highly structured nature of the database (which is used to control the precise form of the rankings that are produced in response to a query). The work discussed below, conversely, is independent of the source or structure of the text and requires only that it has been prepared using a text formatting package that allows the automatic identification of the starts of the paragraphs; the full-text documents used in our study were all prepared using either LaTeX or SROFF.

EXPERIMENTAL DETAILS

Use of INSTRUCT

The basis for our work has been an experimental text retrieval system called INSTRUCT (*IN*teractive *S*ystem for *T*eaching *R*etrieval *U*sing *C*omputational *T*echniques). INSTRUCT was originally developed to demonstrate the principles of nearest neighbour searching to students of librarianship and information science [24–26] but has since been used as a test bed for a number of research projects in various areas of information retrieval[27].

INSTRUCT was designed for the searching of databases of summary records but has been modified for the present study so that it can search sets of paragraphs comprising full-text documents. The facilities used here are as follows (other retrieval mechanisms within INSTRUCT are summarized by Al-Hawamdeh *et al.*[27]).

- Natural language query input. On entering INSTRUCT, the user is asked to submit a query in natural language format: this can be merely a list of keywords or phrases, a sentence-like question or some combination of these. Non-content-bearing words in the query are eliminated using a stopword list and the remaining words then stemmed using an automatic suffix-stripping routine.
- Query modification. The user may delete any of the stems that have been identified, add further keywords as desired or, in a third option, ask the system to identify stems in the database that are similar to words in the query. Two types of similarity measure can be used. The first of these measures the degree of similarity between a query stem and a stem in the database by the number of paragraphs in which they co-occur, since this may help to identify synonyms of the chosen stem. The second measures the similarity by the number of three-character strings, i.e., trigrams, that each stem in the database has in common with the chosen query stem. In both cases, the most similar stems are displayed at the terminal for possible inclusion in the query statement.

- Searching. Having identified the keyword stems that are pertinent to the query, the user may then carry out a best match search. This involves the calculation of weights for each of the query stems. Those used for the initial search are *inverse frequency* weights in which the greatest weights are assigned to those terms in the query which occur least frequently in the database which is to be searched. The presence or absence of the query terms in each of the paragraphs in the full-text document is noted, the sum of the weights for these terms identified, and then the sums of the weights sorted so as to obtain a ranking of the paragraphs. A conventional Boolean searching routine is also provided in which those paragraphs are retrieved that satisfy the logical constraints of a Boolean query in which the query stems are linked by the operators AND, OR and NOT.
- Relevance feedback. The top-ranking paragraphs are then displayed at the terminal for evaluation by the user, who is asked to state whether or not each of them is relevant to the request. After some number of paragraphs, typically ten or twenty, have been inspected in this way, provision is made for the search to be repeated but with the inverse frequency weights being replaced by probabilistic weights[28] that are calculated using the relevance judgements that have been supplied. This re-ranking of the paragraphs thus provides a fully automatic means for query refinement since the weights should more accurately reflect the ability of the query terms to discriminate between the relevant and the non-relevant parts of a full-text document.

Documents and queries

The measurement of retrieval effectiveness requires a *test collection*, i.e., a database for which there is a set of queries together with associated relevance judgements so that it is known which records are relevant to which queries[3,22]. The main problem encountered with building test collections is the need to obtain exhaustive relevance judgements: this requires that each and every record in the database should be considered as a response to each and every query, a totally impracticable task if the database is at all large. We have side-stepped this problem in the work discussed below by careful choice of the full-text documents that are used. Specifically, we have taken coursework essays, journal articles, theses, reports, etc. prepared by students and staff within our Department. The 20 documents that were processed for searching cover a wide range of topics in the general area of information studies including chemical structure databases, the use of cluster analysis, parallel processing techniques, expert systems, information management, and the performance of text retrieval systems. The size and numbers of paragraphs and keyword stems for each document are listed in Table 1.

The authors of the documents were asked to provide sets of questions together with the identities of those paragraphs that are relevant to their questions. In all, 130 natural language queries were obtained, the numbers per document ranging from 4 to 13. Typical queries are as follows:

- What types of hierarchical agglomerative clustering method are available?
- What is the Crandell and Smith algorithm used for?
- What are semantic networks and how are they used in expert systems?
- New information services on chemical patents.
- How easy is it to import LOTUS 1-2-3 files into REFLEX?

Table 1. Statistics of the full-text documents

Document	Paragraphs	Keywords	Size (Kbytes)	Queries
1	261	12837	328	5
2	53	1290	88	6
3	453	21416	612	9
4	297	13573	352	4
5	383	16528	404	5
6	334	17538	508	9
7	246	11913	312	5
8	307	12529	368	6
9	50	3162	80	5
10	62	3584	84	6
11	547	25982	764	8
12	94	4566	112	7
13	107	5154	136	10
14	192	15364	392	7
15	61	1710	96	5
16	204	10993	300	6
17	227	9978	236	4
18	281	7372	196	13
19	101	4512	108	6
20	391	11084	264	4

- Query processing using relevance weighting and threshold values.
- What influence, if any, does pre-library school work have on students' choice of course options and or career expectation?

In some cases, the authors also provided an additional list of keyword descriptors. The use of the author as the source of queries and relevance data is based on the assumptions that he or she is the person best suited to judging the relevance of individual components of that document and that he or she is aware of all such components that are relevant. Both of these assumptions are debatable but their acceptance provides a relatively easy way of obtaining material for a quantitative investigation of the sort considered here. In addition, as is commonly the case in information retrieval research[3], only binary judgements of relevance were obtained, i.e., the paragraphs were labelled as either relevant or non-relevant to each query and no attempt was made to differentiate between their degrees of relevance. All of the searches were carried out by one of us (Suliman Al-Hawamdeh), thus avoiding variations in the level of user search performance.

Evaluation

When a query was matched against a full-text document in a nearest neighbour search, the paragraphs were ranked in decreasing order of similarity and then a cutoff applied to retrieve some fixed number of paragraphs. These were then inspected to ascertain whether or not they were amongst the relevant paragraphs noted by the author; these retrieved, relevant paragraphs were then used for the measurement of retrieval effectiveness.

The most common measures of effectiveness in text retrieval experiments are *recall* and *precision*. Assume that a search retrieves n paragraphs, r of which are relevant, and that there is a total of T relevant paragraphs for that query. Then the recall, R, and the precision, P, are defined to be:

$$R = r/T; \quad P = r/n.$$

Thus, R and P measure the ability of the system to retrieve all the relevant paragraphs and to retrieve only relevant paragraphs, respectively. R and P can conveniently be combined into a single measure, the *effectiveness*, or E, measure of van Rijsbergen[23]. E is a weighted combination of precision and recall where the lower the E value, the greater the effectiveness. This is defined to be:

$$E = 100 \times \left(1 - \frac{(1+\beta^2)PR}{\beta^2 P + R}\right)$$

where β is used to reflect the relative importance to the user of recall and precision ($0 \leq \beta \leq \infty$). $\beta = 1.0$ corresponds to attaching equal importance to precision and to recall, while $\beta = 0.5$ (or 2.0) corresponds to attaching half (or twice) as much importance to recall as to precision. E values were calculated for each query with $\beta = 0.5$, 1.0 and 2.0 and then the mean E values calculated when averaged over the entire set of queries. Note again, that the smaller the E value, the greater the retrieval effectiveness.

COMPARISON OF BOOLEAN AND NEAREST NEIGHBOUR SEARCHING

The first comparative experiments involved carrying out Boolean and nearest neighbour searches on each of the available full-text documents using its set of associated queries. Cutoffs of 10 and 20 paragraphs were used in the nearest neighbour searches while, in the Boolean searches, 10 or 20 paragraphs were selected at random from the output if a particular search formulation retrieved more than the required number. The E values for the nearest neighbour and Boolean searches were compared using the Sign Test as described by van Rijsbergen[23]. A referee pointed out that the use of a fixed cutoff for the number of paragraphs means that the same number of paragraphs is evaluated irrespective of the total number of relevant paragraphs for a particular query; however, this latter number would normally not be available to a user carrying out an actual search and the use of fixed cutoff also simplifies the evaluation of the relevance feedback searches.

The main experimental results are summarized in Table 2. This gives the mean effectiveness values, averaged over all the queries, for the nearest neighbour search and for the Boolean search with cutoffs of 10 and 20 paragraphs. In addition, the table lists the comparable figures for a nearest neighbour search in which relevance feedback is used, i.e., when the relevance judgements on the initial output are used by the system to modify the weights of the query terms as described by Robertson and Sparck Jones[28]. The relevance feedback searches were done in two stages.

- A normal nearest neighbour search was used to identify the top 5 or 10 paragraphs and these were judged for relevance by the user.
- The weights were then recalculated by the system and a further 5 or 10 paragraphs retrieved.

Taking these two parts of the search together gives outputs of size 10 and 20 paragraphs that are then directly comparable to those obtained in the Boolean and in the simple nearest neighbour searches. The summary results of Table 2 are detailed in Tables 3–5. These tables give the mean E values, averaged over the set of queries for each full-text document, when evaluated with β set to 0.5, 1.0 and 2.0. The three E values listed for each value of β correspond to the effectiveness of the nearest neighbour, Boolean and relevance feedback searches (denoted by NN, B and RF, respectively).

Table 2. Mean E values for each search strategy

β	Cutoff-10			Cutoff-20		
	NN	B	RF	NN	B	RF
0.5	71.0	72.3	67.4	80.9	80.5	76.3
1.0	75.3	76.5	71.9	86.2	86.0	82.9
2.0	81.0	82.6	79.5	88.5	87.7	83.4

Table 3. Mean E value calculated for each document with β = 0.5

Document	Cutoff-10			Cutoff-20		
	NN	B	RF	NN	B	RF
1	80.5	84.7	73.8	79.1	88.5	76.9
2	68.3	64.4	72.6	74.5	69.7	69.9
3	77.4	75.1	69.4	85.9	83.8	81.0
4	69.3	70.6	60.6	79.4	75.5	68.8
5	72.4	67.3	65.2	84.8	85.8	78.8
6	79.1	79.1	72.2	86.2	83.1	78.9
7	52.5	65.8	54.4	75.2	75.1	70.4
8	82.7	69.1	79.5	81.1	78.3	84.6
9	70.3	72.4	67.6	76.8	65.5	69.1
10	72.6	72.3	63.8	82.6	84.8	73.4
11	70.5	75.9	63.8	83.6	82.7	67.8
12	61.9	65.6	59.5	76.6	72.9	67.7
13	76.7	77.2	79.0	86.3	82.6	82.2
14	67.7	81.5	64.8	80.0	88.2	74.7
15	68.5	73.7	71.7	82.1	82.5	80.2
16	64.2	68.0	57.9	81.8	83.0	81.7
17	70.2	68.9	73.9	86.6	82.5	80.4
18	72.9	72.4	71.5	85.2	83.3	83.3
19	62.1	60.4	62.4	75.4	74.1	72.9
20	72.4	75.9	64.5	82.5	84.5	68.9

If we consider first the nearest neighbour and Boolean searches, the results in Tables 2–5 suggest that neither of the strategies is noticeably superior to the other. This conclusion is supported by the Sign Test, which shows no statistically significant difference between the two sets of results at the 0.05 level of statistical significance. An analogous lack of difference has been observed in previous searches of summary records on INSTRUCT using these two search strategies[27]. Thus, it may be concluded that nearest neighbour searching provides a viable alternative to Boolean searching for paragraph-based retrieval

Table 4. Mean E value calculated for each document with $\beta = 1.0$

Document	Cutoff-10			Cutoff-20		
	NN	B	RF	NN	B	RF
1	82.5	82.7	62.5	82.5	90.0	85.0
2	80.0	76.3	82.0	81.2	88.1	80.0
3	78.8	72.9	70.0	89.4	81.4	85.6
4	67.5	70.0	65.0	83.7	81.2	76.2
5	82.0	78.0	77.5	91.0	92.0	87.5
6	86.1	85.4	81.4	92.1	90.2	87.1
7	66.0	75.8	68.0	85.0	91.2	82.0
8	89.8	78.8	88.0	89.0	87.0	89.0
9	72.5	74.7	70.0	81.2	75.0	78.3
10	71.6	71.5	70.0	85.8	87.0	72.5
11	72.5	79.7	66.2	87.2	85.3	72.5
12	62.8	67.0	61.4	81.4	78.3	74.2
13	81.1	83.9	85.0	90.5	88.4	86.2
14	73.3	83.6	70.0	86.6	91.7	83.3
15	78.0	84.0	80.0	89.0	88.0	88.0
16	70.0	74.5	65.0	87.5	89.0	89.1
17	80.0	79.1	82.5	92.5	89.5	88.7
18	82.5	81.3	80.0	91.6	90.0	90.4
19	55.0	52.5	48.0	77.5	76.0	77.0
20	75.0	80.0	66.6	87.5	88.4	78.3

Table 5. Mean E value calculated for each document with $\beta = 2.0$

Document	Cutoff-10			Cutoff-20		
	NN	B	RF	NN	B	RF
1	81.5	92.4	83.6	82.9	88.8	91.1
2	88.4	85.9	89.7	95.4	93.4	87.5
3	91.0	83.1	87.6	82.3	87.2	78.4
4	65.9	68.5	66.4	88.0	86.4	82.9
5	89.1	86.5	86.7	95.0	95.7	93.1
6	91.4	89.6	88.5	95.7	94.6	92.7
7	78.2	84.4	79.8	91.5	95.1	89.8
8	94.3	86.6	93.4	93.9	92.7	92.2
9	62.0	66.6	58.2	78.2	83.3	85.5
10	86.4	86.3	88.9	72.3	88.5	78.2
11	72.2	85.8	59.9	89.5	94.5	74.8
12	74.4	62.9	63.8	85.1	72.6	76.2
13	92.0	93.2	88.7	92.5	91.4	87.2
14	78.6	84.2	75.3	91.8	94.7	89.9
15	85.3	90.9	87.1	93.7	90.9	93.1
16	73.1	79.6	69.1	91.7	93.3	93.9
17	87.6	87.0	89.2	95.9	94.1	93.9
18	89.6	88.5	86.7	95.5	94.3	94.8
19	87.2	84.5	84.3	68.4	69.7	78.4
20	73.6	79.8	64.5	91.7	88.5	86.1

from full-text documents, an alternative that is much more appropriate for end-user searching owing to the greater ease of query formulation. In addition, it is worth noting that the E values are comparable in magnitude to those obtained from nearest neighbour searches of databases of summary documents, e.g., the results reported by Griffiths *et al.*[29]. The results from the relevance feedback searches appear to be rather better than both the nearest neighbour and the Boolean searches (since the E values are generally lower than for the other two types of retrieval mechanism); however, use of the Sign Test again reveals no statistically significant differences.

GROUPING OF RELEVANT PARAGRAPHS IN FULL-TEXT DOCUMENTS

There is an alternative approach to nearest neighbour searching in databases of summary records that is very different in concept from that described so far. This approach involves the use of a clustering method to group together documents that have large numbers of keywords in common. A *cluster-based* search then involves matching the query against the resulting clusters of documents, rather than against individual documents, and the documents displayed are those contained within the top-ranked clusters[21,29]. Relevant documents can hence be retrieved, even if they are not very similar to a query, as long as they have been clustered with documents that *are* very similar to the query: indeed, it is possible for relevant documents to be retrieved by this means even if they have not a single term in common with the query.

The full-text environment provides an alternative mechanism for directing the user's attention to relevant paragraphs even if they do not provide a good match with the query statement that is submitted to the system. Specifically, if a paragraph, P_1, is in the same segment of a document as a paragraph, P_2, that is known to be relevant, where 'in the same segment' could correspond, e.g., to being on the same page as P_2 or in the same section of a chapter as P_2, then P_1 is also likely to be relevant and should be retrieved, even if it is not a good match for the query. If this hypothesis is correct, then the retrieval of one or more relevant paragraphs in the same segment of a full-text document provides strong evidence for the retrieval of the other paragraphs in that segment, in addition to those that are highly similar to the query. Thus, in the summary record case, the grouping of documents is a result of the application of some clustering method to the records in the database; in the full-text case considered here, the grouping of paragraphs is a function of their actual location within the body of a text.

The validity of this hypothesis was tested by studying the distribution of the relevant paragraphs for a sample of 25 of the queries associated with full-text documents that had been subdivided by their original authors into subsections, sections and chapters as defined by the $\LaTeX$ text processing package (though some of the documents had only some of these components and there was thus a total of 100, 148 and 174 relevant paragraphs in the subsections, sections and chapters, respectively). The distribution of these relevant paragraphs is summarized in Table 6, from which it is clear that there is a very substantial degree of grouping present. For example, there is one case where no less than six relevant paragraphs are contained within the same subsection of a document and three cases where there are five relevant paragraphs in the same subsection (and still larger numbers in the same section or chapter). It is of interest to compare this observed distribution with that which would be expected if relevant paragraphs were distributed at random, i.e., if there was no tendency for them to occur together. These figures are predicted by the Poisson distribution which states that the probability that a particular document segment contains n relevant paragraphs is given by the equation

$$f(n) = \frac{e^{-\lambda}\lambda^{n}}{n!}$$

where λ is the mean number of relevant paragraphs in that segment (when averaged across the entire set of 25 queries). The calculated values show that not a single subsection or section in a full-text document would be expected to contain more than one relevant paragraph and that only one chapter in such a document would be expected to contain two relevant paragraphs. Thus, these predicted values represent a far smaller degree of grouping than is observed in practice.

Table 6. Distribution of relevant paragraphs within document segments for the sample set of 25 queries

Segment	1	2	3	4	5	6	7	8
Subsection	38	11	5	1	3	1	0	0
Section	31	12	9	5	5	1	1	1
Chapter	23	13	10	7	5	1	4	1

For comparison with these distributions, the retrieval results for the sample set of 25 queries are summarized in a similar manner as shown in Table 7. This table lists the number of times in this query set that one to five relevant paragraphs were retrieved that all came from the same subsection, section or chapter for the three retrieval strategies using a cutoff of 10 paragraphs: there were no occurrences with more than five relevant paragraphs retrieved from the same segment. While these results are obviously inferior to the ‘ideal’ results listed in Table 6, some degree of clustering is still achieved.

Table 7. Distribution of relevant retrieved paragraphs within document segments for the sample set of 25 queries with a cutoff of 10 paragraphs

	NN					B					RF				
Segment	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Subsection	12	4	1	0	0	13	2	1	0	0	15	6	2	0	1
Section	8	7	3	2	0	11	8	4	1	0	10	7	5	2	0
Chapter	7	8	4	2	1	7	9	5	1	1	8	8	5	2	0

The finding that there is some degree of grouping in the output from a nearest neighbour search suggests a novel approach to end-user access to full-text documents. Specifically, we now describe an automatic means of identifying that stretch of continuous text within the body of a full-text document that is likely to contain the greatest amount of relevant material for the initial query that has been put to the system. The procedure starts by carrying out a conventional nearest neighbour search to find the 10 or 20 paragraphs that are most similar to the query. However, the output is not presented directly to the user; instead, the paragraphs are inspected to determine from which segment of the text they come. That segment which occurs most frequently in the set of top-ranking paragraphs is noted and this segment is then presented to the user, *in toto*, as the output from the search. Thus, rather than presenting a number of paragraphs from several different segments of the document that is being searched, the user is presented with a single segment of continuous text. This segment is the one that contains the largest number of individual paragraphs that are strongly correlated with the query; accordingly, from the results in Tables 6 and 7, it is also expected to contain other relevant paragraphs that are not so similar. In effect, given an initial query statement, this retrieval mechanism guides the user to that part of the document where he or she should start reading, thus avoiding the need to scan through large screen-fulls of text to identify individual, top-ranking paragraphs as in a conventional nearest neighbour search of a full-text document.

This idea was tested using the same set of 25 queries as previously, with the E values being calculated for the total number of paragraphs in the subsection, section or chapter identified by analysis of the top 10 or top 20 paragraphs from the nearest neighbour search. The results of these searches are listed in Table 8, together with the simple nearest neighbour searches. The general trends evident in these figures are confirmed by use of the Sign Test which reveals that:

- the presentation of a single subsection is significantly more effective, at the 0.05 level of significance, than the presentation of either the top 10 or the top 20 paragraphs from a conventional nearest neighbour search;
- the presentation of a single section is significantly more effective than the presentation of the top 20 paragraphs; and
- the presentation of a single chapter is significantly less effective than the presentation of either the top 10 or the top 20 paragraphs.

Table 8. Mean E values for the sample set of 25 queries for the display of a single sub-section, section or chapter, as compared with a normal NN search

	Cutoff-10				Cutoff-20			
β	Subsection	Section	Chapter	NN	Subsection	Section	Chapter	NN
0.5	62.8	77.4	86.9	73.6	57.7	69.2	85.1	79.4
1.0	70.3	81.7	91.7	78.5	65.1	74.2	90.7	84.9
2.0	77.0	85.2	95.2	82.1	73.8	78.2	94.6	88.9

CONCLUSIONS

In this paper, we have considered strategies for the retrieval of individual paragraphs from full-text documents. Tests with 20 such documents and 130 queries and the associated relevance judgements show that there is no significant difference between the performances of nearest neighbour, Boolean and relevance feedback searches; since the Boolean search requires a much greater degree of user expertise for query formulation, these results support the use of ranking search strategies for end-user access to full-text databases. We have also demonstrated that analysis of the top-ranking paragraphs in the output from a nearest neighbour search can assist a user in deciding which part of a full-text document should be inspected first when it is displayed at a terminal.

The results that have been obtained to date suggest several avenues for further investigation. Perhaps most importantly, the work here has taken little account of the weights that are assigned to the terms used to characterize the queries and the paragraphs that are being searched. As noted in the subsection on Use of INSTRUCT, the query terms have been weighted using inverse frequency weighting[22]; this is simple in concept and effective in operation but there are many other types of scheme that can be used to weight query terms[21]. Moreover, the document paragraphs here are described only by the presence or absence of terms, without consideration of how frequently they occur within a paragraph, and studies of summary document retrieval suggest that the use of such information can be used further to increase the effectiveness of nearest neighbour searching[21,23]. An investigation of a range of weighting schemes for full-text searching is reported by Al-Hawamdeh and Willett[30]. Again, the identification of a relevant paragraph could be used to provide browsing facilities by using that paragraph as the source of query terms for a second search, thus providing an alternative type of relevance feedback mechanism; studies of browsing in summary databases suggest that such facilities are very popular with users. It would also be of interest to compare the results obtained in the section on Grouping of Relevant Paragraphs in Full-text Documents, where a particular segment of text is identified automatically, with those obtainable from manual scanning of tables of contents, which provide a more conventional way of accessing a full-text. Finally, one can imagine different ways of processing the output from a nearest neighbour search; e.g., one might take account of the sizes of different segments when deciding which one should be displayed.

ACKNOWLEDGEMENTS

We thank the British Council for funding, Stephen J. Wade for assistance with INSTRUCT, and the members of our department, past and present, who have provided the full-text documents and relevance judgements that have been used in our experiments.

REFERENCES

1. J.H. Ashford and P. Willett, *Text Retrieval and Document Databases*, Chartwell-Bratt, Lund 1988.
2. J.E. Rowley, ‘An overview of microcomputer text retrieval packages’, *Aslib Proceedings*, **40**, 311–319 (1988).
3. G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York 1983.
4. M.E. Williams, ‘Electronic databases’, *Science*, **228**, 445–456 (1985).
5. D.C. Blair and M.E. Maron, ‘An evaluation of retrieval effectiveness for a full-text document retrieval system’, *Communications of the ACM*, **28**, 289–299 (1985).
6. D.C. Blair, ‘Full text retrieval: evaluation and implications’, *International Classification*, **13**, 18–23 (1986).
7. J.S. Ro, ‘An evaluation of the applicability of ranking algorithms to improve the effectiveness of full-text retrieval. I. On the effectiveness of full-text retrieval’, *Journal of the American Society for Information Science*, **39**, 73–78 (1988).
8. J.S. Ro, ‘An evaluation of the applicability of ranking algorithms to improve the effectiveness

of full-text retrieval. II. On the effectiveness of ranking algorithms on full-text retrieval', *Journal of the American Society for Information Science*, **39**, 147–160 (1988).

9. C. Tenopir, 'Full text databases', *Annual Review of Information Science and Technology*, **19**, 215–246 (1984).
10. C. Tenopir, 'Full text database retrieval performance', *Online Review*, **9**, 149–164 (1985).
11. R. Wagers, 'The decision to search databases full text', *Proceedings of the Tenth International Online Information Meeting*, 93–107 (1986).
12. D.P. Dabney, 'The curse of Thamos: an analysis of full-text legal document retrieval', *Law Library Journal*, **78**, 5–40 (1986).
13. J. O'Connor, 'Retrieval of answer-sentences and answer-figures from papers by text searching', *Information Processing and Management*, **11**, 155–164 (1975).
14. J. O'Connor, 'Answer passage retrieval by text searching', *Journal of the American Society for Information Science*, **31**, 227–239 (1981).
15. J.F. Cove and B.C. Walsh, 'Online text retrieval via browsing', *Information Processing and Management*, **24**, 31–37 (1988).
16. L.M. Bernstein and R.E. Williamson, 'Testing of a natural language retrieval system for a full text knowledge base', *Journal of the American Society for Information Science*, **35**, 235–247 (1984).
17. C.R. Watters, M.A. Shepherd, E.W. Grundke and P. Brodorik, 'Integration of menu retrieval and Boolean retrieval from a full-text database', *Online Review*, **9**, 391–402 (1985).
18. S.M. Cohen, C.A. Schermer and L.R. Garson, 'Experimental program for online access to ACS primary documents', *Journal of Chemical Information and Computer Sciences*, **20**, 247–252 (1980).
19. C. Tenopir, 'Search strategies for full text databases', *Proceedings of the ASIS Annual Meeting*, **25**, 80–86 (1988).
20. D. Lucarella, 'A search strategy for large document bases', *Electronic Publishing*, **1**, 105–116 (1988).
21. G. Salton, *Automatic Text Processing: the Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley, Reading MA, 1989.
22. P. Willett, ed., *Document Retrieval Systems*, Taylor Graham, London, 1988.
23. C.J. van Rijsbergen, *Information Retrieval*, 2nd. edition, Butterworth, London, 1979.
24. I.G. Hendry, P. Willett and F.E. Wood, 'INSTRUCT: a teaching package for experimental methods in information retrieval. Part 1. The user's view', *Program*, **20**, 245–263 (1986).
25. I.G. Hendry, P. Willett and F.E. Wood, 'INSTRUCT: a teaching package for experimental methods in information retrieval. Part 2. Computational aspects', *Program*, **20**, 382–393 (1986).
26. S.J. Wade and P. Willett, 'INSTRUCT: a teaching package for experimental methods in information retrieval. Part 3. Browsing, clustering and query expansion', *Program*, **22**, 44–61 (1988).
27. S. Al-Hawamdeh, D. Ellis, K.C. Mohan, S.J. Wade and P. Willett, 'Best match document retrieval: development and use of INSTRUCT', *Proceedings of the Twelfth International Online Information Meeting*, 761–777 (1988).
28. S.E. Robertson and K. Sparck Jones, 'Relevance weighting of search terms', *Journal of the American Society for Information Science*, **27**, 129–146 (1976).
29. A. Griffiths, H.C. Luckhurst and P. Willett, 'Using inter-document similarity in document retrieval systems', *Journal of the American Society for Information Science*, **37**, 3–11 (1986).
30. S. Al-Hawamdeh and P. Willett, 'Comparison of index term weighting schemes for the ranking of paragraphs in full-text documents', *International Journal of Information and Library Research*, in press.