In the Proceedings of SIGIR '93, Pittsburgh, PA

Subtopic Structuring for Full-Length Document Access

Marti A. Hearst
Computer Science Division
571 Evans Hall, UC Berkeley
Berkeley, CA 94720 USA
and Xerox Palo Alto Research Center
marti@cs.berkeley.edu

Christian Plaunt
Library and Information Studies
207c SLIS, UC Berkeley
Berkeley, CA 94720 USA
chris@bliss.berkeley.edu

Abstract

We argue that the advent of large volumes of full-length text, as opposed to short texts like abstracts and newswire, should be accompanied by corresponding new approaches to information access. Toward this end, we discuss the merits of imposing *structure* on full-length text documents; that is, a partition of the text into coherent multi-paragraph units that represent the pattern of subtopics that comprise the text. Using this structure, we can make a distinction between the main topics, which occur throughout the length of the text, and the subtopics, which are of only limited extent. We discuss why recognition of subtopic structure is important and how, to some degree of accuracy, it can be found. We describe a new way of specifying queries on full-length documents and then describe an experiment in which making use of the recognition of local structure achieves better results on a typical information retrieval task than does a standard IR measure.

1 Introduction

Full-length documents have only recently become available online in large quantities, although technical abstracts and short newswire texts have been accessible for many years (Tenopir & Ro 1990). For this reason, most information retrieval methods are better suited for accessing abstracts than longer documents. In this paper, we argue that the advent of full-length text should be accompanied by corresponding new approaches to information access.

Abstracts are compact and information-dense. Most of the (non-stopword) terms in an abstract are salient for retrieval purposes because they act as placeholders for multiple occurrences of those terms in the original text, and because generally these terms pertain to the most important topics in the text. Consequently, if the text is of any sizeable length, it will contain many subtopic discussions that are never mentioned in its abstract (if it has one).

For these reasons, when a user submits a vector-space query against a collection of abstracts, the user is in effect specifying that the system find documents whose combination of main topics is most like that of the query. In other words, when abstracts are compared via the vector-space model, they are positioned in a multi-dimensional space where the closer two abstracts are to one another, the more topics they are presumed to have in common. This is often reasonable because when comparing abstracts, the goal is to discover which pairs of documents are most alike. For example, a query against a set of medical abstracts which contains terms for the name of a disease, its symptoms, and possible treatments is best matched against an abstract with as similar a constitution as possible.

Furthermore, most IR similarity measures treat the terms in a document uniformly throughout. That is, a term's weight is the same no matter where it occurs in the text.[1] Many researchers assume this is a valid assumption when working with abstracts, since it is a fair approximation to say that the location of the term does not significantly effect its import (although (Liddy 1991) discusses the usefulness of understanding the structure of an abstract when using a natural-language based IR approach). These comments apply as well to short news articles, another text type commonly studied in information retrieval research.

A problem with applying traditional information retrieval methods to full-length text documents[2] is that

[1] Small windows of adjacency information are sometimes used in boolean systems, but not in probabilistic or vector-space models.

[2] For the discussions in this paper, the term "full-length document" refers to an unabstracted expository text which can be of any length but a typical example would be a five-page science magazine article or a 20 page environmental impact report. It excludes documents composed of short "news bites" or any other disjointed, although lengthy, text. We also assume that the document does not have detailed orthographically marked structure;

the structure of full-length documents is quite different from that of abstracts. One way to view an expository text is as a sequence of subtopics set against a "backdrop" of one or two main topics. A long text will be comprised of many different subtopics which may be related to one another and to the backdrop in many different ways. The main topics of a text are discussed in its abstract, if one exists, but subtopics usually are not mentioned. Therefore, instead of querying against the entire content of a document, a user should be able to issue a query about a coherent subpart, or subtopic, of a full-length document, and that subtopic should be specifiable with respect to the document's main topic(s).

In the remainder of this paper we discuss why recognition of subtopic structure is important and how, to some degree of accuracy, it can be found. We describe a new way of specifying queries on full-length documents and speculate about how subtopic structuring can improve on categorization and index selection tasks. We then describe an experiment in which making use of the recognition of local structure achieves better results on a typical information retrieval task than does a standard measure. A closing discussion follows.

2 Subtopic Structure

2.1 Why to find Subtopic Structure

Consider a *Discover* magazine article about the Magellan space probe's exploration of Venus. A reader divided this 23-paragraph article into the following segments with the labels shown, where the numbers indicate paragraph numbers:

1- 2	*Intro to Magellan space probe*
3- 4	*Intro to Venus*
5- 7	*Lack of craters*
8-11	*Evidence of volcanic action*
12-15	*River Styx*
16-18	*Crustal spreading*
19-21	*Recent volcanism*
22-23	*Future of Magellan*

Assume that the topic of "volcanic activity", or perhaps "geological activity", is of interest to a user. Crucial to a system's decision to retrieve this document is the knowledge that a dense discussion of volcanic activity, rather than a passing reference, appears. Since volcanism is not one of the text's two main topics, we shouldn't expect the number of references to this term to dominate the statistics of a vector space model. On the other hand, we don't necessarily want to select a document just because there are a few references to the target terms.

(Croft *et al.* 1990) describes work that takes advantage of this kind of information.

The goal should be to determine whether or not a *relevant* discussion of a concept or topic appears. A simple approach to distinguishing between a true discussion and a passing reference is to determine the "locality" of the references. In the computer science operating systems literature, "locality" refers to the fact that over time memory access patterns tend to concentrate in localized clusters, rather than being distributed evenly throughout memory. Similarly, in full-length texts, if a set of references to a particular concept occur in close proximity to one another, this is a good indicator of topicality. For example, the term *volcanism* occurs 5 times in the Magellan article, the first four instances of which occur in 4 adjacent paragraphs, along with accompanying discussion. In contrast, the term *scientists*, which is not a valid subtopic, occurs 13 times, distributed somewhat evenly throughout. By its very nature, a subtopic will not be discussed throughout an entire text. Similarly, true subtopics are not indicated by only passing references. The term *belly dancer* occurs only once, and its related terms are confined to the one sentence it appears in. As its usage is only a passing reference, "belly dancing" is not a true subtopic of this text.

Our solution to the problem of retaining valid subtopical discussions while at the same time avoiding being fooled by passing references is to make use of locality information and partition documents according to their subtopical structure. This approach's capacity for improving a standard information retrieval task is verified in the experiments discussed in Section 4.

2.2 How to find Subtopic Structure

Using Orthographically Marked Segments

One way to get an approximation to subtopic structure is to break the document into paragraphs, or for very long documents, sections. In both cases this entails using the orthographic marking supplied by the author to determine topic boundaries. Salton and Buckley (Salton & Buckley 1991a),(Salton & Buckley 1991b) have done the most comprehensive work to date on issues pertaining to full-length text. They have compared paragraphs within a large document (e.g., Salton's book), articles within an online encyclopedia, and electronic mail messages (inquiries and their replies). According to Salton and Buckley, a good way to ensure that two larger segments, such as two paragraphs, are similar is to make sure they are similar both overall and locally (via comparing sentence-by-sentence). For two sections to be similar, they must be similar overall, at the paragraph level, and at the sentence level. Their results show that this procedure is more effective than using overall information alone. To accommodate for the fact that most paragraphs differ in length, they normalize the term frequency component for the comparisons.

In the applications they've described, Salton and Buckley focus on finding subparts of a large document that co-refer or are very similar in content (or co-referring texts in a corpus, as is the case in the electronic mail messages experiment) for the purposes of hypertext, for example. Their focus is on how to find similarity among blocks of text of greatly differing length, and not so much on the role of the text block in the document that it is a part of.

(Ro 1988a) has performed experiments addressing the issue of retrieval from full texts in contrast to using controlled vocabulary, abstracts, and paragraphs alone. Performing boolean retrieval for a set of nine queries against business management journal articles, Ro found that retrieving against full text produced the highest recall but the lowest precision of all the methods. In subsequent experiments, (Ro 1988b) tried various weighting schemes in an attempt to show that retrieving against full text would perform better than against paragraphs alone, but did not achieve significant results to this effect. These experiments are not entirely relevant to the efforts described here because of their focus on the boolean paradigm.

Using Even-Sized Blocks

Another way to approximate local structure in long documents is to hack the documents into even-size pieces, without regard for any boundaries. (Stanfill & Waltz 1992) report on such a technique, using the efficiency of a massively parallel computer. They divide the documents into 30-word segments and compare the queries to each segment. They also combine the scores for adjacent 30-word segments in case they break the document in a very inopportune place, and then report the best n combined scores. This simple method, performed on texts consisting of newswires, magazines, newspapers, among others, achieves good results after extensive testing (the authors cite a precision-recall product of 0.65 on their task).

An explanation for why this technique is so effective is that it takes advantage of localized discussions. In our experiments, however, using unmotivated segments worked less well overall than using paragraphs or motivated segments (see Section 4).

TextTiling

We are interested in exploring the performance of *motivated* segmentation, i.e., segmentation that reflects the text's true underlying subtopic structure, which often spans paragraph boundaries.

(Hahn 1990) has eloquently addressed the need for imposing structure on full-length documents in order to improve information retrieval, but proposes a knowledge-intensive, strongly domain dependent approach, which is difficult to scale to sizeable text collections. In contrast, we want an algorithm that can be implemented and tested on a large, diverse text collection.

Toward this end, we have developed TextTiling, a method for partitioning full-length text documents into coherent multi-paragraph units (Hearst 1993b). TextTiling approximates the subtopic structure of a document by using patterns of lexical connectivity to find coherent subdiscussions. The layout of the 'tiles' is meant to reflect the pattern of subtopics contained in an expository text. The approach uses quantitative lexical analyses to determine the extent of the tiles and to classify them with respect to a general knowledge base. The tiles have been found to correspond well to human judgements of the major subtopic boundaries of science magazine articles.

The algorithm is a two step process; first, all pairs of adjacent blocks of text (where blocks are usually 3-5 sentences long) are compared and assigned a similarity value, and then the resulting sequence of similarity values, after being graphed and smoothed, is examined for peaks and valleys. High similarity values, implying that the adjacent blocks cohere well, tend to form peaks, whereas low similarity values, indicating a potential boundary between tiles, create valleys. Figure 1 shows such a graph for the magazine article mentioned in Section 1. The vertical lines indicate where human judges thought the topic boundaries should be placed.

The one adjustable parameter is the size of the block used for comparison. This value, labeled k, varies slightly from text to text; as a heuristic it is assigned the average paragraph length (in sentences), although the block size that best matches the human judgement data is sometimes one sentence greater or fewer. Actual paragraphs are not used because their lengths can be highly irregular, leading to unbalanced comparisons.

Similarity is measured by putting a twist on the tf.idf measurement (Salton 1988). In standard tf.idf, terms that are frequent in an individual document but relatively infrequent throughout the corpus are considered to be good distinguishers of the contents of the individual document. In TextTiling, each block of k sentences

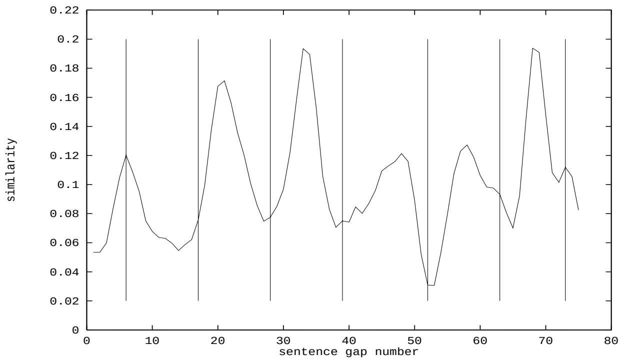

Figure 1: Results of TextTiling a 77-sentence popular science article. Vertical lines indicate actual topic boundaries as determined by human judges, and the graph indicates computed similarity of adjacent blocks of text. Peaks indicate coherency, and valleys indicate potential breaks between tiles.

is treated as a unit unto itself, and the frequency of a term within each block is compared to its frequency in the entire document.[3] This helps bring out a distinction between local and global extent of terms; if a term is discussed frequently but within a localized cluster (thus indicating a cohesive passage), then it will be weighted more heavily than if it appears frequently but scattered evenly throughout the entire document, or infrequently within one block. Thus if adjacent blocks share many terms, and those shared terms are weighted heavily, there is strong evidence that the adjacent blocks cohere with one another.

Similarity between blocks is calculated by a cosine measure: given two text blocks $b1$ and $b2$,

$$cos(b1, b2) = \frac{\sum_{t=1}^{n} w_{t,b1} w_{t,b2}}{\sqrt{\sum_{t=1}^{n} w_{t,b1}^2 \sum_{t=1}^{n} w_{t,b2}^2}}$$

where t ranges over all the terms in the document and $w_{t,b1}$ is the tf.idf weight assigned to term t in block $b1$. Thus if the similarity score between two blocks is high, then not only do the blocks have terms in common, but the terms they have in common are relatively rare with respect to the rest of the document. The evidence in the reverse is not as conclusive: if adjacent blocks have a low similarity measure, this does not necessarily mean they don't cohere; however, in practice this negative evidence is often justified.

The graph is then smoothed using a discrete convolution[4] of the similarity function with the function $h_k(\cdot)$, where:

$$h_k(i) \equiv \begin{cases} \frac{1}{k^2}(k - |i|), & |i| \leq k - 1 \\ 0, & otherwise \end{cases}$$

The result is smoothed further with a simple median smoothing algorithm (Rabiner & Schafer 1978), with a window of size three, to eliminate small local minima. Tile boundaries are determined by locating the lowermost portions of valleys in the resulting plot. The actual values of the similarity measures are not taken into account; the relative differences are what are of consequence. This algorithm is fully implemented and is employed to segment full-length texts in the experiments described in Section 4.

In the next section, we discuss a new information access paradigm that makes use of the kind of structure found by TextTiling.

3 A New Kind of Query

We wish to present the user with the capacity to specify queries that are appropriate for full length documents. What a user should really be able to do is distinguish a search for a subtopic from a search for a main topic. Furthermore, the user should be able to specify a search for a subtopic *with respect to* some main topic.

As mentioned above, the Salton and Buckley algorithm combines overall and local information about a unit of text in order to make a similarity judgement. We are suggesting that this kind of information be kept distinct in order to facilitate a retrieval paradigm in which a user can specify not only the subtopic to retrieve on, but also which main topic(s) the subtopic should appear in the context of. Toward this end, after a document is TextTiled, an index is built consisting of two parts: the global, main topic index, and a set of local, subtopic indexes.

To see why this distinction might be useful, consider the following scenario: A user would like to find a discussion of funding for cold fusion research. There is a long text about cold fusion that has a two-paragraph discussion of funding two-thirds of the way in. This discussion, because it is in the context of a document about cold fusion, does not mention the term *cold fusion* anywhere near the discussion of funding. A full-document retrieval will either assign low rank to this document because funding-related terms are infrequent relative to the whole, or else it will assign high rank to *any* articles about cold fusion. A retrieve against individual paragraphs will either assign low rank to this document because it will see only funding terms but no cold fusion terms in the relevant segment, or it will give high rank to *any* documents that have discussions of funding.

[3] The algorithm uses a large "stop list"; i.e., closed class words and other very frequent terms are omitted from the calculation.

[4] The authors are grateful to Michael Braverman for proving that the smoothing algorithm is equivalent to this convolution.

If instead we have a bipartite representation of the document's contents, as described above, then we should be able to accommodate this query. We would retrieve all documents that have been determined to have a main topic of "cold fusion" and a subtopic of "funding". This would also provide a mechanism for highlighting the relevant part of the document. The reverse query could be specified as well; documents with a main topic of "funding policies" with subtopics on "cold fusion" might be appropriate as well.

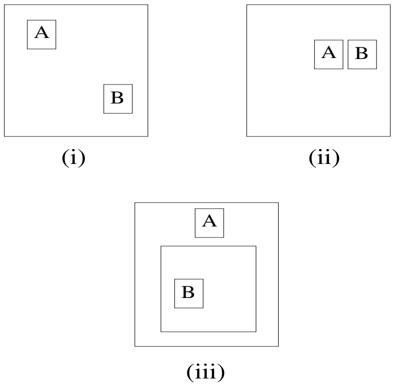

Figure 2: The Subordination Query. If a user would like to find a document that discusses topic B in the context of topic A, a retrieval method based on the conjunction of terms either retrieves any document containing both terms, whether or not they modify one another (i), or only those documents that have both terms adjacent to one another, or otherwise explicitly modifying one another (ii). If, on the other hand, we know that a main topic of the text is A, then if a subtopic B occurs somewhere distant from A, we can still assume a modificational relationship (iii).

Note that issuing a query for a subtopic in the context of a main topic should be considered to be qualitatively different from issuing a conjunction. A conjunction should specify either a join of two or more main topics or a join of two or more subtopics – it should imply conjoining two like items. In contrast, "in the context of" can be thought of as a subordinating relation, like a relative clause that restricts the scope of the noun phrase that it modifies. (See Figure 2.) This notion of specifying a query in which one topic is subordinate to another to our knowledge has not previously been suggested for information retrieval.

Another way to make use of the bipartite representation of the contents of a full-length document is to invoke a capacity for "change of background". If we think of the main topic information as being separate from the subtopic information, we can specify similarity search on subtopic structure alone. We could actually strip out the terms associated with the main topic from all of the subtopic segments, yielding a "topic-neutral" subtopic discussion. Then topic-neutral subtopics can be compared, presumably yielding improved results. E.g., the discussion of funding in the cold fusion text could be matched to a discussion of funding in a text about superconductivity, free from the conflicts that would result if the main topic terms were intermixed with the subtopic terms.

These ideas are still in the experimental stage and have not yet been implemented. (Hearst 1993a) discusses in more detail innovative ways to use this bipartite indexing strategy.

4 Using Subtopic Structure to Improve Retrieval: An Experiment

Our approach to retrieval should reflect our assumption that full-length text is meaningfully different in structure from abstracts and short articles. Below we demonstrate that taking text structure into account can produce better results than using full-length documents in the standard way. By working within this paradigm, we've developed an approach to vector-space based retrieval that appears to work better than retrieving against entire documents or against segments or paragraphs alone.

Instead, a query is matched against motivated segments, and then the scores from the top segments for each document are summed. The highest resulting sums indicate which documents should be retrieved. In our test set, this method produced higher precision and recall than retrieving against entire documents or against segments or paragraphs alone.

4.1 Method

For these experiments, we used the Ziff subset of the TIPSTER test collection.[5] For the purposes of our experiments, this subset possesses two very important and hard-to-find features: somewhat lengthy, full-length texts and pre-determined relevance judgments for a set of queries.[6]

[5] The authors would like to thank Donna Harman who made this collection available to us as members of the Berkeley Full-Text Retrieval Research Group, a participant in the DARPA-sponsored TREC conference. Note that at the time of writing the results of the TREC conference are not available in publishable form and so we do not reference them.

[6] Our use of the term "query" corresponds to what TIPSTER calls a "topic". A list of queries and documents used in the experiments is available from the authors.

All of our indexing and retrieval was carried out using Salton's SMART system (version 7.0). The experiment was composed of the following steps:

1. Break each text into segments (i.e., run TextTiling)
2. Index each segment as a separate document (while maintaining a pointer to its "parent" text)
3. Retrieve against the segmented document collection
4. Recombine the retrieved segments by parent text retaining only a predetermined number of the top-scoring texts
5. Compare the results with standard query/full document retrieval

In the experimental runs themselves, we followed the example of (Salton & Buckley 1991a) by retrieving a threshold number of documents for each query; we used cutoffs of 5, 10, 15, 20, 25, and 30 retrieved documents.

Though similar to standard vector-space retrieval, there are few notable differences here. After the texts are segmented, they are indexed not as 274 full-length documents, but as 5,926 individual segment-length documents. This allows terms used in a dense subtopic discussions, which are now appearing as *independent* short documents, to be indexed and weighted more accurately in relation to the global context than if they had been a small part of a larger text.

We experimented with several ways to retrieve against the segments. The most successful of these was to sum the similarities of the top 200 segments retrieved. For example, for one query (number 11), the following segments were the top 10 retrieved:

<NUM>	4620	<TEXT>	224	<SEGMENT>	1	108.980583
<NUM>	4623	<TEXT>	224	<SEGMENT>	4	61.340954
<NUM>	4642	<TEXT>	225	<SEGMENT>	9	61.312168
<NUM>	369	<TEXT>	25	<SEGMENT>	1	57.008450
<NUM>	4628	<TEXT>	224	<SEGMENT>	9	54.120296
<NUM>	370	<TEXT>	25	<SEGMENT>	2	53.048084
<NUM>	387	<TEXT>	25	<SEGMENT>	19	51.410706
<NUM>	4633	<TEXT>	224	<SEGMENT>	14	47.891445
<NUM>	4626	<TEXT>	224	<SEGMENT>	7	46.451294
<NUM>	4627	<TEXT>	224	<SEGMENT>	8	46.153339

These were combined by summing the query-to-document similarity of those segments which came from the same document. In the above example, segment numbers 4620, 4623, 4628, 4633, 4626 and 4627 all originated in text 224, and so were added together, as were 369, 370 and 387, to form the following set:

<TEXT>	224	364.937911
<TEXT>	25	161.467240
<TEXT>	225	61.312168

In this case, text 224 is the only relevant text for query 11, which is clearly reflected in the scores of the summed retrieved set. Seven more document sums would have to be found in order to have retrieved the top 10 documents.

In order to compare the effect of the summing with using single segments alone, we performed two other tests using the segments. The first test is 'fff" (for "first few found"). In fff, if the current threshold is 20, that is, the top 20 documents are to be retained, if the top 20 segments are from 20 different documents, then all 20 are included in the precision/recall evaluation. However, if a document has multiple representatives, only the highest scoring representative is preserved in the final tally. Applied to the retrieved ten segments above, this method produced the following set:

<TEXT>	224	108.980583
<TEXT>	225	61.312168
<TEXT>	25	57.008450

Note that even in this short example the order and magnitude of the similarity values differ depending on the method used. With larger retrieval thresholds, the difference becomes even more noticeable.

In the second method, called "fud" (for "first unique documents"), again only one segment is allowed to represent a document, but this time the algorithm keeps searching down the ranked list of segments until a set of segments whose size equals the threshold size is found. In the example above, the algorithm would continue until 7 more appropriate segments are found. Of fff and fud, fff is expected to have better precision and worse recall, whereas fud is expected to have better recall and worse precision.

Finally, for comparison to standard retrieval performance, we indexed the same document collection, but in its unsegmented, original form (referred to as "full"), and ran the same queries, retrieving the top number of texts up to the predetermined threshold.

The next two subsections describe the experiment in more detail, followed by a detailed description of the results.

4.2 Preparation

The Test Data. As mentioned above, we used the Ziff part of the Tipster collection, because it's texts were the best examples of full-length documents that we could find in a test collection with pre-determined relevance judgements.[7] This allowed us to select and

[7] Actually, many of the texts were not of the desired form, consisting of short, disjointed "news bites".

process a subset of 43 queries and 274 documents based on a combination of (i) relevance – each query has one or more relevant texts and each text has one or more queries for which it has been judged relevant – and (ii) length – the text part of the document had to have at least 1,500 words.

The Queries. Once the queries and text of interest were identified and extracted, they were substantially cleaned up. Queries were cleaned automatically with a small program that removed various formatting information, "stock" phrases that appeared in every query and blank lines. Explicit "NOT" clauses (e.g., "NOT product upgrades") were eliminated if they were the first content-bearing phrase on a line, otherwise, they were ignored. All other potentially content-bearing information, including domain designations, description, definitions, etc., were left untouched. In nearly all cases, this cut about 40 words from a query.

A sample query after cleaning is:

```
companies
that develop multimedia applications,
standards, or specifications.
multimedia, multi-media, interactive media
CD-ROM, desktop video
applications, developers
standards, interfaces
Multimedia - Communicating information
in more than one form and it
includes the use of text, audio, graphics
and full- motion video.
```

The Texts. The documents themselves were in a pseudo-sgml format which included a good deal of information besides the running text, e.g. author, title, journal title, descriptors, document numbers, etc., and so were handled in a manner similar to the queries. From the raw formatted documents, we extracted and saved only the actual running text (as marked by pseudo-sgml) and the unique document number, nothing else. With the aid of several small programs, we removed the remaining pseudo-sgml markers (e.g. "@amp;"), fixed up sentence and paragraph boundaries, etc. This process reduced the size of the texts by about 12% on average.

With this done, the queries and texts were ready for segmenting and the retrieval experiments. The average segment length was approximately 164 words, the average paragraph length was approximately 53 words, and the average full text length was 3,557 words.

4.3 Weighting Schemes

For our experiments, we used two variants of tf.idf term weighting schemes. For full length documents, following the suggestion of (Salton & Buckley 1991a) we used then "enhanced" *atc* weights for query/document similarity comparison. We also used this weight for indexing and calculating inter-segment similarity. The *atc* measure normalizes the weights for document length, giving all documents an equal chance for retrieval:

$$w_{ik} = \frac{(0.5 + 0.5\frac{f_{ik}}{\max f_{ip}})(\log \frac{N}{n_p})}{\sqrt{\sum_{k=1}^{t}(0.5 + 0.5\frac{f_{ik}}{\max f_{ip}})^2(\log \frac{N}{n_p})^2}}$$

where w_{ik} is the weight assigned to term T_k in document D_i; f_{ik} is the frequency of T_k in D_i, $\max f_{ip}$ is the maximum term frequency of any term in this vector, N is the number of documents in the collection, t is the number of unique terms in the collection, and n_p is the number of documents that contain T_k.

On segmented documents, we used the "unenhanced" *ntn* measure,

$$w_{ik} \;=\; f_{ik} \;*\; log\frac{N}{n_p}$$

for determining similarity between queries and segments. *ntn* differs from *atc* in two ways. First, it performs no manipulation on the term frequency part of the equation, and since our segments were of fairly regular length (compared to how widely unprocessed documents can vary), we did not require this normalization. Second, since we were retrieving the top N documents, as opposed to documents whose score was above a certain similarity value threshold, we did not require the scores to be normalized between .5 and 1, as *atc* does.

4.4 Results

The results of the experiments can be summarized as follows. For each of the 43 queries and at each of the predetermined document count cutoff values, we ran the following tests:

- "full" (standard) query/document retrieval, retaining all top ranked texts up to the threshold
- "fff" segment retrieval, retain only the top texts found associated with the top segments retrieved
- "fud" segment retrieval, retaining each of the unique texts found associated with the retrieved segments, only stopping when the threshold of retrieved texts is reached

- "sum"
  segment retrieval, combining same-document segment/query similarities until the threshold of texts has been reached

We found, as shown in Figure 3, that by indexing full-text documents at the level of motivated segments, performing retrieval against these segments, and summing the results of these sub-document retrievals, we could increase both recall and precision over conventional query/document retrieval consistently and substantially. Our experiments obtained improvements of from 18.9% to 28.2% (for both precision and recall) over normally indexed documents. This indicates that combining the scores of the highest segmenting highly ranked segments of a document is an effective means for performing a standard similarity retrieval against full-length documents.

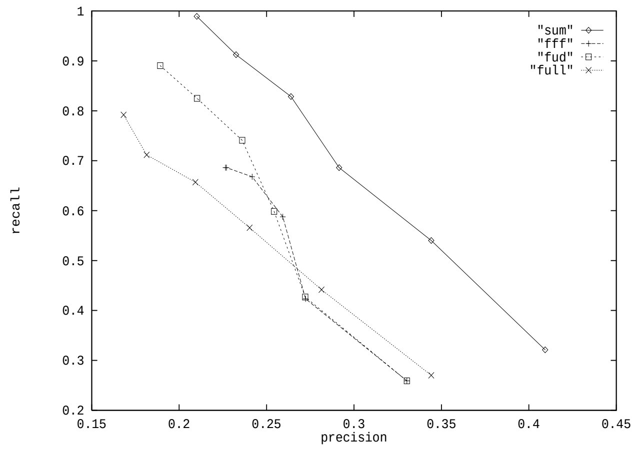

Figure 3: Comparison of summing segments to the "full", "fud", and "fff" methods for queries against the long documents in the Tipster Ziff collection. Summing the scores of the highest ranked segments was most effective for both precision and recall.

Overall, we saw the following improvements when moving from full text retrieval to the "sum" method, for both precision and recall (the fact that they are the same is an artifact of using the same number of queries and texts):

Top 5: 18.9%
Top 10: 23.3%
Top 15: 21.3%
Top 20: 26.1%
Top 25: 28.2%
Top 30: 24.9%

We also found that indexing by segment (or paragraph) without summing, in both the "fff" and "fud" versions, also showed improvements over the conventional method, as shown in Figure 3. The plots in figures 3, 4, and 5 show precision vs. recall for the six cutoff values, which are marked as points on the graphs. For each method, the leftmost point marks the cutoff at 30 documents, the point to its right marks the cutoff at 25, and so on. All of the results presented in these plots also appear in Table 1.

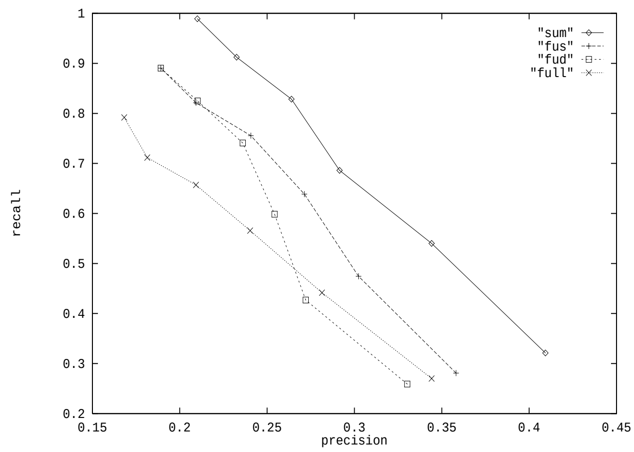

Figure 4: Comparison of summing segments against "full", "fud", and "fus", in order to show that the non-adjaceny information in the summing approach appears to add useful information.

We were curious to know whether or not the improvements resulted from the fact that the summing procedure combined adjacent segments into larger relevant groups, or whether the fact that similarity values from segments from nonadjacent parts of the text were contributing to the results. We then made up a new test set, which we call "fus" (first unique super-segment) to compare against "sum". In "fus", if two segments from the same document are in the top 200 segments retrieved for a query and these segments are adjacent in the original document, then they are combined into one superseg- ment that spans both segments. We expected fus to do better than the individual-segment methods "fud" and "fff" and not to do as well as sum, since it would not be making use of information from distant parts of the document. Figure 4 supports our expectations.

However, paragraphs produced very similar results to segments throughout these experiments, although we expect segments to outperform paragraphs as the segmentation procedure improves. In Figure 5 we compare summing using paragraphs against summing using our motivated segments. Partitioning the texts into evenly sized blocks, without regard for orthographic boundaries, performed less well than using motivated segments or paragraphs in the experiments so far.

Despite the fact that "sum" performs consistently better than each of these methods on both precision and recall at each measurement point, we did not find the differences in their means to be statistically significant (t-test, $p < 0.05$, holding recall fixed and testing precision) on this test collection. Nevertheless, we find these results to be quite encouraging and plan to pursue variations on the summing approach in the near future, perhaps incorporating sophisticated evidence combination methods, such as those explored in (Turtle 1991), instead of simple summing.

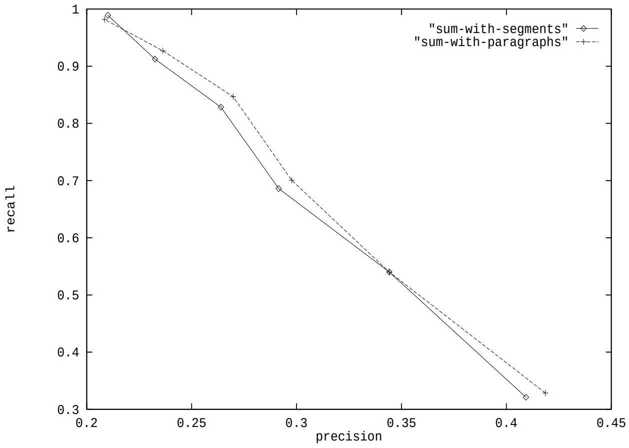

Figure 5: Comparison of summing segments using paragraphs vs. using motivated segments.

5 Conclusions

We have discussed the importance of recognizing the structure of full-length text for the purposes of information retrieval, emphasizing that most existing similarity-based techniques implicitly assume that the documents being queried against are of uniform structure and import. We suggested partitioning full-length documents into multi-paragraph units as an approximation their underlying subtopic structure. We then suggested a new paradigm for information access queries based on this structure, focusing on a distinction between terms that correspond to the main topics of the text and those that are part of subtopic discussions only; specifically we suggest allowing users to specify a subordinating relation between main topic and subtopics. This should be especially useful when the local context of the subtopic does not contain references to main topic terms.

We also performed an experiment that demonstrates the utility of treating full-length documents as being composed of a sequence of locally concentrated discussions. The idea is to divide the documents into moti-

	total retrieved	total relevant	precision	recall
Top 5				
full	215	74	0.34	0.27
fff	215	71	0.33	0.25
fud	215	71	0.33	0.25
fus	215	77	0.36	0.28
sum	215	88	0.40	0.32
par	215	90	0.42	0.33
Top 10				
full	430	121	0.28	0.44
fff	426	116	0.27	0.42
fud	430	117	0.27	0.42
fus	430	130	0.30	0.47
sum	430	148	0.34	0.54
par	430	148	0.34	0.54
Top 15				
full	645	155	0.24	0.56
fff	621	161	0.25	0.58
fud	645	164	0.25	0.59
fus	645	175	0.27	0.64
sum	645	188	0.29	0.68
par	645	192	0.30	0.70
Top 20				
full	860	180	0.20	0.65
fff	757	183	0.24	0.66
fud	860	203	0.23	0.74
fus	860	207	0.24	0.76
sum	860	227	0.26	0.82
par	860	232	0.27	0.85
Top 25				
full	1075	195	0.18	0.71
fff	828	188	0.22	0.68
fud	1075	226	0.21	0.82
fus	1075	225	0.21	0.82
sum	1075	250	0.23	0.91
par	1075	254	0.24	0.93
Top 30				
full	1290	217	0.16	0.79
fff	830	188	0.22	0.68
fud	1290	244	0.18	0.89
fus	1290	244	0.19	0.89
sum	1290	271	0.21	0.98
par	1290	269	0.21	0.98

Table 1: Comparison of the methods at each cutoff value. "par" refers to summing using paragraphs as opposed to segments.

vated segments, retrieve the top-scoring segments that most closely match the query, and then sum the scores for all segments that are from the same document. This causes the most relevant portions of the documents to contribute to the final score for the document. It could be the case that for queries that touch on both a main topic and a subtopic of a document, this procedure approximates the main topic/subtopic symbiosis; this is a subject for further exploration. Also to be explored is the question of what portions of the documents contribute to the sum – are they several different discussions about the same subtopic, or different pieces of the text corresponding to different parts of the query?

We also plan to explore how to determine what the main topic terms and the subtopic terms are, after the segmentation has taken place, and pursue the idea of separating main topic terms from the subtopic segments they intermix with, in order to facilitate "topic-neutral" queries. We plan to use a category-based lexical disambiguation algorithm based on that of (Yarowsky 1992) for the purpose of assigning groups of terms to higher level categories, to facilitate disambiguated term expansion within subtopic segments.

Acknowledgments

The authors would like to thank Bill Cooper and Ray Larson for their help and encouragement, and David Lewis and two anonymous reviewers for their suggestions of improvements to this paper. The first author's research was sponsored in part by the University of California and Digital Equipment Corporation under Digital's flagship research project Sequoia 2000: Large Capacity Object Servers to Support Global Change Research.

References

Croft, W. Bruce, Robert Krovetz, & H. Turtle. 1990. Interactive retrieval of complex documents. *Information Processing and Management*, 26(5):593–616.

Hahn, Udo. 1990. Topic parsing: Accounting for text macro structures in full-text analysis. *Information Processing and Management*, 26(1):135–170.

Hearst, Marti A. 1993a. Cases as structured indexes for full-length documents. In *Proceedings of the 1993 AAAI Spring Symposium on Case-based Reasoning and Information Retrieval*, Stanford,CA.

Hearst, Marti A. 1993b. TextTiling: A quantitative approach to discourse segmentation. Technical Report 93/24, Sequoia 2000 Technical Report, University of California, Berkeley.

Liddy, Elizabeth. 1991. The discourse level structure of empirical abstracts – an exploratory study. *Information Processing and Management*, 27(1):55–81.

Rabiner, Lawrence R. & Ronald W. Schafer. 1978. *Digital processing of speech signals.* New Jersey: Prentice-Hall, Inc.

Ro, Jung Soon. 1988a. 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(2):73–78.

Ro, Jung Soon. 1988b. 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(3):147–160.

Salton, Gerard. 1988. *Automatic text processing : the transformation, analysis, and retrieval of information by computer.* Reading, MA: Addison-Wesley.

Salton, Gerard & Chris Buckley. 1991a. Automatic text structuring and retrieval: Experiments in automatic encyclopedia searching. In *Proceedings of SIGIR*, 21–31.

Salton, Gerard & Chris Buckley. 1991b. Global text matching for information retrieval. *Science*, 253:1012–1015.

Stanfill, Craig & David L. Waltz. 1992. Statistical methods, artificial intelligence, and information retrieval. In *Text-based intelligent systems: Current research and practice in information extraction and retrieval*, ed. by Paul S. Jacobs, 215–226. Lawrence Erlbaum Associates.

Tenopir, Carol & Jung Soon Ro. 1990. *Full text databases.* New Directions in Information Management. Greenwood Press.

Turtle, Howard. 1991. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222.

Yarowsky, David. 1992. Word sense disambiguation using statistical models of roget's categories trained on large corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, 454–460, Nantes, France.