To Appear in *Computational Linguistics*, March 1997

TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages

Marti A. Hearst*
Xerox PARC

Abstract

TextTiling is a technique for subdividing texts into multi-paragraph units that represent passages, or subtopics. The discourse cues for identifying major subtopic shifts are patterns of lexical co-occurrence and distribution. The algorithm is fully implemented and is shown to produce segmentation that corresponds well to human judgments of the subtopic boundaries of twelve texts. Multi-paragraph subtopic segmentation should be useful for many text analysis tasks, including information retrieval and summarization.

1. Introduction

Most work in discourse processing, both theoretical and computational, has focused on analysis of inter-clausal or inter-sentential phenomena. This level of analysis is important for many discourse processing tasks, such as anaphor resolution and dialogue generation. However, important and interesting discourse phenomena also occur at the level of the paragraph. This article describes a paragraph-level model of discourse structure based on the notion of *subtopic shift*, and an algorithm for subdividing expository texts into multi-paragraph "passages" or *subtopic segments*.

In this work, the structure of an expository text is characterized as a sequence of subtopical discussions that occur in the context of one or more main topic discussions. Consider a 21-paragraph science news article, called *Stargazers*, whose main topic is the existence of life on earth and other planets. Its contents can be described as consisting of the following subtopic discussions (numbers indicate paragraphs):

1–3	*Intro – the search for life in space*
4–5	*The moon's chemical composition*
6–8	*How early earth-moon proximity shaped the moon*
9–12	*How the moon helped life evolve on earth*
13	*Improbability of the earth-moon system*
14–16	*Binary/trinary star systems make life unlikely*
17–18	*The low probability of non-binary/trinary systems*
19–20	*Properties of earth's sun that facilitate life*
21	*Summary*

Subtopic structure is sometimes marked in technical texts by headings and subheadings. Brown and Yule (1983, 140) state that this kind of division is one of the most basic in discourse. However, there are many expository texts that consist of long sequences of

* 3333 Coyote Hill Rd, Palo Alto, CA, USA, 94304, hearst@parc.xerox.com

paragraphs with very little structural demarcation, and for these a subtopical segmentation can be useful.

This article describes fully implemented techniques for the automatic detection of multi-paragraph subtopical structure. Because the goal is to partition texts into contiguous, nonoverlapping subtopic segments, I call the general approach TextTiling (Hearst, 1993; Hearst, 1994a; Hearst, 1994b).[1] Subtopic discussions are assumed to occur within the scope of one or more overarching main topics, which span the length of the text. This two-level structure is chosen for reasons of computational feasibility and for the purposes of the application types described below.

TextTiling makes use of patterns of lexical co-occurrence and distribution. The algorithm has three parts: tokenization into terms and sentence-sized units, determination of a score for each sentence-size unit, and detection of the subtopic boundaries, which are assumed to occur at the largest valleys in the graph that results from plotting sentence-units against scores. Three methods for score assignment have been explored: *blocks, vocabulary introductions*, and *chains*, although only the first two are evaluated in this article (the third is discussed in (Hearst, 1994b)). All three scoring methods make use only of patterns of lexical co-occurrence and distribution within texts, eschewing other kinds of discourse cues.

The ultimate goal of passage-level structuring is not just to identify the subtopic units, but also to identify and label their subject matter. This article focuses only on the discovery of the segment boundaries, but there is extensive ongoing research on automated topic classification (Lewis and Hayes, 1994). Most classification work focuses on identifying main topic(s), as opposed to TextTiling's method of finding both globally distributed main topics and locally occurring subtopics; nevertheless, variations on some existing algorithms should be applicable to subtopic classification.

The next section argues for the need for algorithms that can detect multi-paragraph subtopic structure (referred to here interchangeably as passages and sub:topic segments), and discusses application areas that should benefit from such structure. Section 3 describes in more detail what is meant in this article by "subtopic" and presents a description of the discourse model that underlies this work. Section 4 introduces the general framework of using lexical co-occurrence information for detecting subtopic shift, and describes other related work in empirical discourse analysis. The TextTiling algorithms are described in more detail in Section 5 and their performance is assessed in Section 6. Finally, Section 7 summarizes the work and describes future directions.

2. Why Multi-Paragraph Units?

In school we are taught that paragraphs are to be written as coherent, self-contained units, complete with topic sentence and summary sentence. In real-world text, these expectations are often not met. Paragraph markings are not always used to indicate a change in discussion, but instead can sometimes be invoked just to break up the physical appearance of the text in order to aid reading (Stark, 1988). A conspicuous example of this practice can be found in the layout of the columns of text in many newspapers (Longacre, 1979). Brown and Yule (1983, pp. 95-96) note that text genre has a strong influence on the role of paragraph markings, and that markings differ for different languages. Hinds (1979, p. 137) also suggests that different discourse types have different organizing principles.

1 A free version of the code, written in C, is available for research purposes. Contact the author for more information.

Although most discourse segmentation work is done at a finer granularity than that suggested here, multi-paragraph segmentation has many potential applications. TextTiling is geared towards expository text; that is, text that explicitly explains or teaches, as opposed to, say, literary texts, since expository text is better suited to the main target applications of information retrieval and summarization. More specifically, TextTiling is meant to apply to expository text that is not heavily stylized or structured, and for simplicity does not make use of heading or other kinds of orthographic information. A typical example is a five-page science magazine article or a twenty page environmental impact report.

This section concentrates on two application areas for which the need for multi-paragraph units has been recognized: hypertext display and information retrieval. There are also potential applications in some other areas, such as text summarization. Some summarization algorithms extract sentences directly from the text. These methods make use of information about the relative positions of the sentences in the text (Kupiec, Pedersen, and Chen, 1995; Chen and Withgott, 1992). However, these methods do not use subtopic structure to guide their choices, focusing more on the beginning and ending of the document and on position within paragraphs. Paice (1990) recognizes the need for taking topical structure into account but does not suggest a method for determining such structure.

Another area that models the multi-paragraph unit is automated text generation. Mooney, Carberry, and McCoy (1990) present a method centered around the notion of Basic Blocks: multi-paragraph units of text, each of which consists of (1) an organizational focus such as a person or a location, and (2) a set of concepts related to that focus. Their scheme emphasizes the importance of organizing the high level structure of a text according to its topical content, and afterwards incorporating the necessary related information, as reflected in discourse cues, in a finer-grained pass.

2.1 Online Text Display and Hypertext

Research in hypertext and text display has produced hypotheses about how textual information should be displayed to users. One study of an online documentation system (Girill, 1991) compares display of fine-grained portions of text (i.e., sentences), full texts, and intermediate sized units. Girill finds that divisions at the fine-grained level are less efficient to manage and less effective in delivering useful answers than intermediate sized units of text.

The author does not make a commitment about exactly how large the desired text unit should be, but talks about "passages" and describes passages in terms of the communicative goals they accomplish (e.g., a problem statement, an illustrative example, an enumerated list). The implication is that the proper unit is the one that groups together the information that performs some communicative function; in most cases this unit will range from one to several paragraphs. (Girill also finds that using document boundaries is more useful than ignoring document boundaries, as is done in some hypertext systems, and that pre-marked sectional information, if available and not too long, is an appropriate unit for display.)

Tombaugh, Lickorish, and Wright (1987) explore issues relating to ease of readability of long texts on CRT screens. Their study explores the usefulness of multiple windows for organizing the contents of long texts, hypothesizing that providing readers with spatial cues about the location of portions of previously read texts will aid in their recall of the information and their ability to quickly locate information that has already been read once. In the experiment, the text is divided using pre-marked sectional information, and one section is placed in each window. They conclude that segmenting the text by means of multiple windows can be very helpful if readers are familiar with the mechanisms

supplied for manipulating the display.

Converting text to hypertext in what is called post-hoc authoring (Marchionini, Liebscher, and Lin, 1991) requires division of the original text into meaningful units (a task noted by these authors to be a challenging one) as well as meaningful interconnection of the units. Automated multi-paragraph segmentation should help with the first step of this process, and is more important than ever now that pre-existing documents are being put up for display on the World Wide Web. Salton et al. (1996) have recognized the need for multi-paragraph units in the automatic creation of hypertext links as well as theme generation, (this work is discussed in Section 5).

2.2 Information Retrieval

In the field of information retrieval there has recently been a surge of interest in the role of passages in full text. Until very recently most information retrieval experiments made use only of titles and abstracts, bibliographic entries, or very short newswire articles, as opposed to full text. When long texts are available there arises the question: can retrieval results be improved if the query is compared against only a passage or subpart of the text, as opposed to the text as a whole? And if so, what size unit should be used? In this context, "passage" refers to any segment of text isolated from the full text. This includes author-determined segments, marked orthographically (paragraphs, sections, and chapters) (Hearst and Plaunt, 1993; Salton, Allan, and Buckley, 1993; Moffat et al., 1994) and/or automatically derived units of text, including fixed-length blocks (Hearst and Plaunt, 1993; Callan, 1994), segments motivated by subtopic structure (TextTiles) (Hearst and Plaunt, 1993), or segments motivated by properties of the query (Mittendorf and Schäuble, 1994).

Hearst and Plaunt (1993), in some early passage-based retrieval experiments, report improved results using passages over full-text documents, but do not find a significant difference between using motivated subtopic segments and arbitrarily chosen block lengths that approximated the average subtopic segment length. Salton, Allan, and Buckley (1993), working with encyclopedia text, find that comparing a query against orthographically marked sections and then paragraphs is more successful than comparing against full documents alone.

Moffat et al. (1994) find, somewhat surprisingly, that manually supplied sectioning information may lead to poorer retrieval results than techniques that automatically subdivide the text. They compare two methods of subdividing long texts. The first consists of using author-supplied sectioning information. The second uses a heuristic in which small numbers of paragraphs are grouped together until they exceed a size threshold. The results are that the small, artificial multi-paragraph groupings seemed to perform better than the author-supplied sectioning information (which usually consisted of many more paragraphs than their subdivision algorithm or TextTiling would create). More experiments are necessary in this vein to firmly establish this result, but it does lend support to the conjecture that multi-paragraph subtopic-sized segments, such as those produced by TextTiling, are useful for similarity-based comparisons in information retrieval.

It will not be all that surprising if motivated subtopic segments are not found to perform significantly better than appropriately sized, but arbitrarily segmented, units in a coarse-grained information retrieval evaluation. At TREC, the most prominent information retrieval evaluation platform (Harman, 1993), the top 1000 documents are evaluated for each query, and the best-performing systems tend to use very simple statistical methods for ranking documents. In this kind of evaluation methodology, subtle distinctions in analysis techniques tend to be lost, whether those distinctions be how accurately words are reduced to their roots (Hull and Grefenstette, 1995; Harman, 1991), or exactly how passages are subdivided. The results of Hearst and Plaunt (1993), Salton, Allan, and

Buckley (1993) and Moffat et al. (1994) suggest that it is the nature of the intermediate size of the passages that matters.

Perhaps a more appropriate use of motivated segment information is in the display of information to the user. One obvious way to use segmentation information is to have the system to display the passages with the closest similarity to the query, and to display a passage-based summary of the documents' contents.

As a more elaborate example of using segmentation in full text information access, I have used the results of TextTiling in a new paradigm for display of retrieval results (Hearst, 1995). This approach, called *TileBars*, allows the user to make informed decisions about which documents and which passages of those documents to view, based on the distributional behavior of the query terms in the documents. TileBars allows users to specify different sets of query terms, as discussed later. The goal is to simultaneously and compactly indicate

> (i) the relative length of the document,
> (ii) the frequency of the term sets in the document, and
> (iii) the distribution of the term sets with respect to the document and to each other.

TextTiling is used to partition each document, in advance, into a set of multi-paragraph subtopical segments.

Figure 1 shows an example query about automated systems for medical diagnosis, run over the ZIFF portion of the TIPSTER collection (Harman, 1993). Each large rectangle next to a title indicates a document, and each square within the rectangle represents a TextTile in the document. The darker the tile, the more frequent the term (white indicates 0, black indicates 8 or more hits; the frequencies of all the terms within a term set are added together). The top row of each rectangle corresponds to the hits for Term Set 1, the middle row to hits of Term Set 2, and the bottom row to hits for Term Set 3. The first column of each rectangle corresponds to the first TextTile of the document, the second column to the second TextTile, and so on. The patterns of gray-level are meant to provide a compact summary of which passages of the document matched which topics of the query.

Users' queries are written as lists of words, where each list, or *term set*, is meant to correspond to a different component of the query.[2] This list of words is then translated into conjunctive normal form. For example, the query in the Figure is translated by the system as: (*patient* OR *medicine* OR *medical*) AND (*test* OR *scan* OR *cure* OR *diagnosis*) AND (*software* OR *program*). This formulation allows the interface to reflect each conceptual part of the query: the medical terms, the diagnosis terms, and the software terms. The document whose title begins *"VA automation means faster admissions"* is quite likely to be relevant to the query, and has hits on all three term sets throughout the document. By contrast, the document whose title begins *"It's hard to ghostbust a network ..."* is about computer-aided diagnosis, but has only a passing reference to *medical* diagnosis, as can be seen by the graphical representation.

This version of the TileBars interface allows the user to filter the retrieved documents according to which aspects of the query are most important. For example, if the user decides that medical terms should be better represented, the "min hits" or "min distribution" constraint on this term set can be adjusted accordingly. Min hits indicates the minimum number of times words from a term set must appear in the document in order for

2 This query format was found to be unproblematic for users in a separate study (Hearst et al., 1996), and is also used in the Grateful Med medical information system (Hersh et al., 1995).

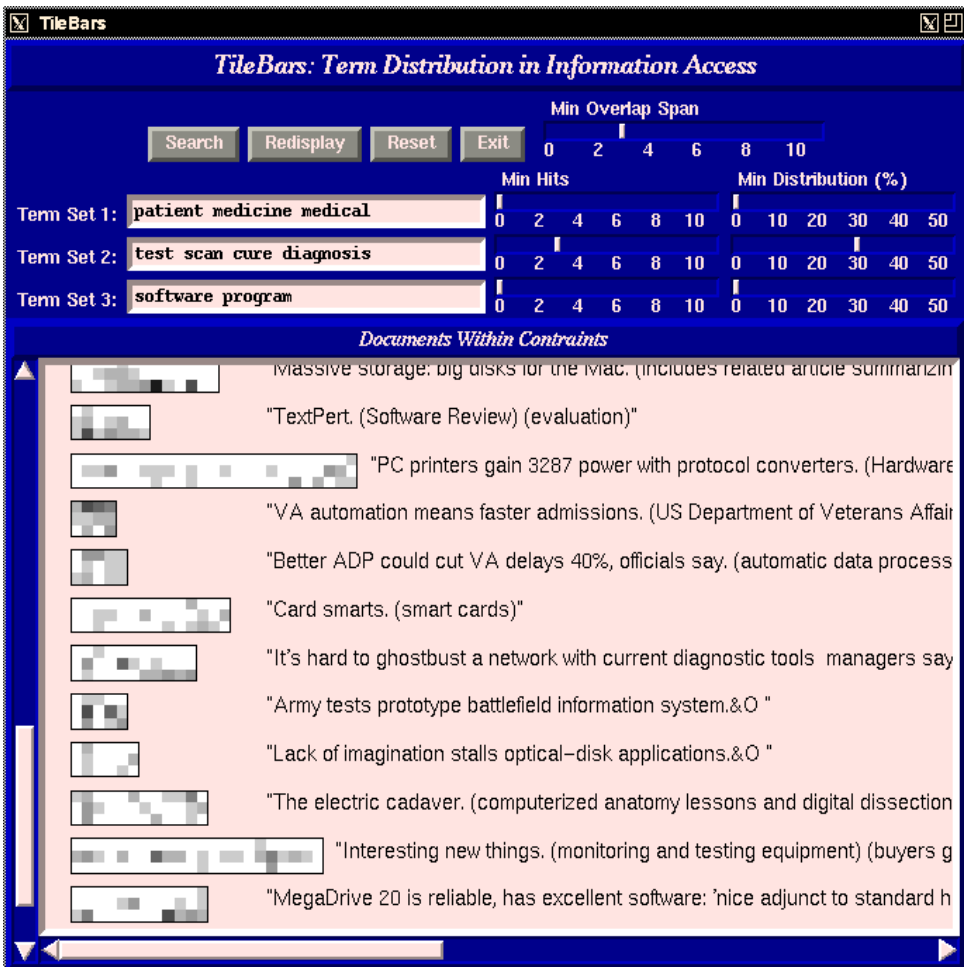

Figure 1
The TileBar Display on a query about automated systems for medical diagnosis.

it to be displayed. Similarly, min distribution indicates the minimum percentage of tiles that must have a representative from the term set. The setting "min overlap span" refers to the minimum number of tiles that must have at least one hit from each of the three term sets. In Figure 1, the user has indicated that the diagnosis aspect of the query must be strongly present in the retrieved documents, by setting the min distribution to 30% for the second term set.[3]

When the user mouse-clicks on a square in a TileBar, the corresponding document is displayed beginning at the selected TextTile. Thus the user can also view the subtopic structure within the document itself.

This section has discussed why multi-paragraph segmentation is important and how it might be used. The next section elaborates on what is meant by multi-paragraph subtopic structure, casting the problem in terms of detection of topic or subtopic *shift*.

3. Coarse-Grained Subtopic Structure

3.1 What is Subtopic Structure?

In order to describe the detection of subtopic structure, it is important to define the phenomenon of interest. The use of the term *subtopic* here is meant to signify pieces of text "about" something and is not to be confused with the topic/comment distinction (Grimes, 1975), also known as the given/new contrast (Kuno, 1972), found within individual sentences.

The difficulty of defining the notion of *topic* is discussed at length in (Brown and Yule, 1983)(§3). They note (pp. 69-70):

> The notion of 'topic' is clearly an intuitively satisfactory way of describing the unifying principle which makes one stretch of discourse 'about' something and the next stretch 'about' something else, for it is appealed to very frequently in the discourse analysis literature.
>
> Yet the basis for the identification of 'topic' is rarely made explicit.

After many pages of attempting to pin the concept down, they suggest, as one alternative, investigating *topic-shift* markers instead (pp. 94-95):

> It has been suggested … that instead of undertaking the difficult task of attempting to define 'what a topic is', we should concentrate on describing what we recognize as **topic shift**. That is, between two contiguous pieces of discourse which are intuitively considered to have two different 'topics', there should be a point at which the shift from one topic to the next is marked. If we can characterize this marking of topic-shift, then we shall have found a structural basis for dividing up stretches of discourse into a series of smaller units, each on a separate topic. … The burden of analysis is consequently transferred to identifying the formal markers of topic-shift in discourse.

This notion of looking for a shift in content bears a close resemblance to Chafe's notion of The Flow Model of discourse in narrative texts (Chafe, 1979), in description of which he writes (pp 179-180):

3 Most likely this setting information is too complicated for a typical user; I have performed some experiments to determine how to set these constraints automatically (Hearst, 1996) to be used in future versions of the interface.

> Our data … suggest that as a speaker moves from focus to focus (or from thought to thought) there are certain points at which there may be a more or less radical change in space, time, character configuration, event structure, or, even, world. … At points where all of these change in a maximal way, an episode boundary is strongly present. But often one or another will change considerably while others will change less radically, and all kinds of varied interactions between these several factors are possible.[4]

Thus, rather than identifying topics (or subtopics) *per se*, several theoretical discourse analysts have suggested that changes or shifts in topic can be more readily identified and discussed. TextTiling adopts this stance. The problem remains, then, of how to detect subtopic shift. Brown and Yule (1983) consider in detail two markers: adverbial clauses and certain kinds of prosodic markers. By contrast, the next subsection will show that lexical co-occurrence patterns can be used to identify subtopic shift.

3.2 Relationship to Segmentation in Hierarchical Discourse Models

Much of the current work in empirical discourse processing makes use of hierarchical discourse models, and several prominent theories of discourse assume a hierarchical segmentation model. Foremost among these are the attentional/intentional structure of Grosz and Sidner (1986) and the Rhetorical Structure Theory of Mann and Thompson (1987). The building blocks for these theories are phrasal or clausal units, and the targets of the analyses are usually very short texts, typically one to three paragraphs in length.[5] Many problems in discourse analysis, such as dialogue generation and turn-taking (Moore and Pollack, 1992; Walker and Whittaker, 1990), require fine-grained, hierarchical models that are concerned with utterance-level segmentation. Progress is being made in the automatic detection of boundaries at this level of granularity using machine learning techniques combined with a variety of well-chosen discourse cues (Litman and Passonneau, 1995).

By contrast, TextTiling has the goal of identifying major subtopic boundaries, attempting only a linear segmentation. We should expect to see, in grouping together paragraph-sized units instead of utterances, a decrease in the complexity of the feature set and algorithm needed. The work described here makes use only of lexical distribution information, in lieu of prosodic cues such as intonational pitch, pause, and duration (Hirschberg and Nakatani, 1996), discourse markers such as *oh, well, ok, however* (Schiffrin, 1987; Litman and Passonneau, 1995), pronoun reference resolution (Passonneau and Litman, 1993; Webber, 1988) and tense and aspect (Webber, 1987; Hwang and Schubert, 1992). From a computational viewpoint, deducing textual topic structure from lexical occurrence information alone is appealing, both because it is easy to compute, and because discourse cues are sometimes misleading with respect to the topic structure (Brown and Yule, 1983)(§3).

4. Detecting Subtopic Change via Lexical Co-occurrence Patterns

TextTiling assumes that a set of lexical items is in use during the course of a given subtopic discussion, and when that subtopic changes, a significant proportion of the vocabulary

4 Interestingly, Chafe arrived at the Flow Model after working extensively with, and then becoming dissatisfied with, a hierarchical model of paragraph structure like that of Longacre (1979).

5 Discourse work at the multi-paragraph level has been mainly in the theoretical, as opposed to computational, realm, notably the work on macrostructures (van Dijk, 1980; van Dijk, 1981) and story grammars (Lakoff, 1972; Rumelhart, 1975).

changes as well. The algorithm is designed to recognize episode boundaries by determining where thematic components like those listed by Chafe (1979) change in a maximal way. However, unlike other researchers who have studied setting, time, characters, and the other thematic factors that Chafe mentions, I attempt to determine where a relatively large set of active themes changes simultaneously, regardless of the *type* of thematic factor. This is especially important in expository text in which the subject matter tends to structure the discourse more so than characters, setting, and so on. For example, in the *Stargazers* text introduced in Section 1, a discussion of "continental movement", "shoreline acreage", and "habitability" gives way to a discussion of "binary and unary star systems". This is not so much a change in setting or character as a change in subject matter.

The flow of subtopic structure as determined by lexical co-occurrence is illustrated graphically in Figure 2. This figure shows the distribution, by sentence number, of selected terms from the *Stargazers* text. The number of times a given word occurs in a given sentence is shown, with blank spaces indicating zero occurrences. Words that occur frequently throughout the text (e.g., *life, moon*) are often indicative of the main topic(s) of the text. Words that are less frequent but more uniform in distribution, such as *form* and *scientist*, tend to be neutral and do not provide much information about the divisions within the discussions. The remaining words are what are of interest here. They are "clumped" together, and it is these clumps or groups that TextTiling assumes are indicative of the subtopic structure. The problem of segmentation therefore becomes the problem of detecting where these clumps begin and end.

For example, words *binary* through *planet* have considerable overlap in sentences 58 – 78, and correspond to the subtopic discussion "Binary/trinary star systems make life unlikely" shown in the (manually produced) outline in Section 1. There is also a well-demarcated cluster of terms between sentences 35 and 50, corresponding to the grouping together of paragraphs 10, 11, and 12 by human judges who have read the text, and to the subtopic discussion in Section 1 of "How the moon helped life evolve on earth".

These observations suggest that a very simple take on lexical cohesion relations (Halliday and Hasan, 1976) can be used to determine subtopic boundaries. However, from the diagram it is evident that simply looking for chains of repeated terms (as suggested by Morris and Hirst (1991)) is not sufficient for determining subtopic breaks. Even combining terms that are closely related semantically into single chains is insufficient, since often several different themes are active within the same segment. For example, sentences 37 – 51 contain dense interactions among the terms *move, continent, shoreline, time, species,* and *life*, and all but the latter occur only in this region. (It is, however, the case that the interlinked terms of sentences 57 – 71, *space, star, binary, trinary, astronomer, orbit*, are closely related semantically, assuming the appropriate senses of the words.)

Because groups of words that are not necessarily closely related conceptually seem to work together to indicate subtopic structure, I adopt a technique that can take into account the occurrences of multiple simultaneous themes rather than use chains of lexical cohesion relations alone. This viewpoint is also advocated by Skorochod'ko (1972), who suggests discovering a text's structure by dividing it up into sentences and seeing how much word overlap appears among the sentences. The overlap forms a kind of intra-structure; fully connected graphs might indicate dense discussions of a topic, while long spindly chains of connectivity might indicate a sequential account. The central idea is that of defining the structure of a text as a function of the connectivity patterns of the terms that comprise it, in contrast with segmentation guided primarily by fine-grained discourse cues such as register change and cue words.

Many researchers, e.g., (Halliday and Hasan, 1976; Tannen, 1989; Walker, 1992), have

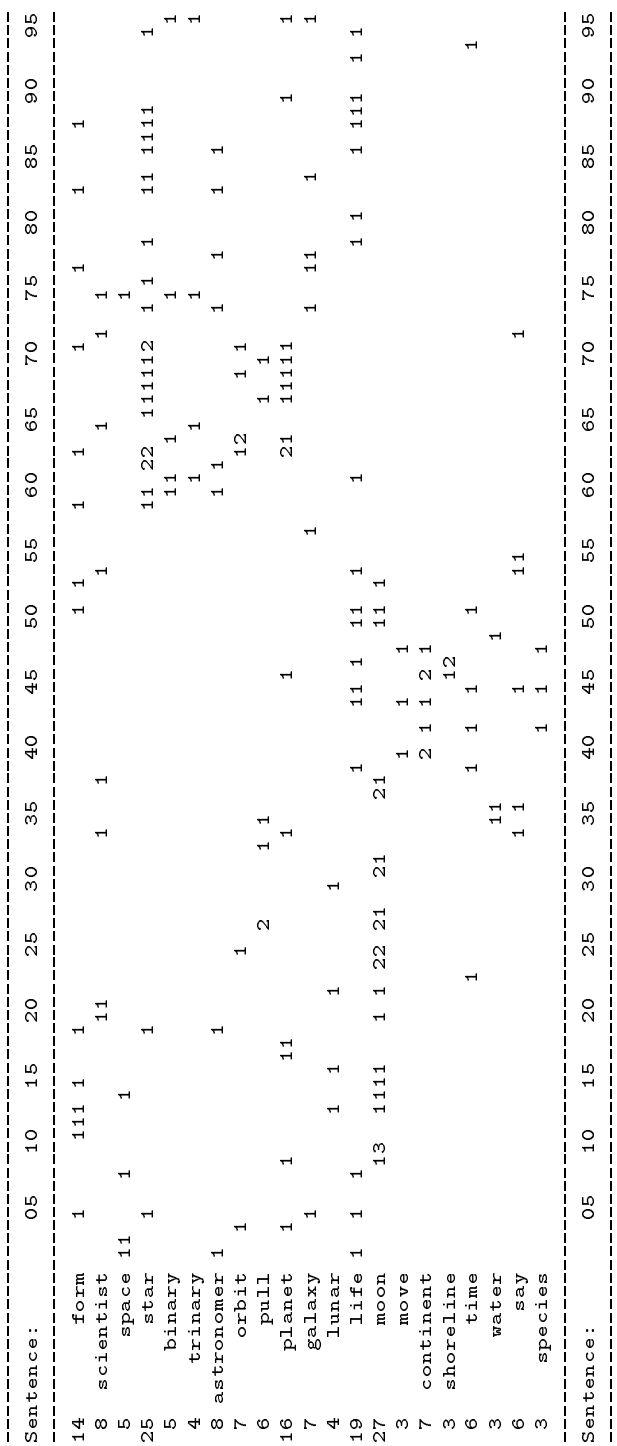

Figure 2
Distribution of selected terms from the *Stargazer* text, with a single digit frequency per sentence number (blanks indicate a frequency of zero).

noted that term repetition is a strong cohesion indicator. Phillips (1985) suggests performing "an analysis of the distribution of the selected text elements relative to each other in some suitable text interval ... for whatever patterns of association they may contract with each other as a function of repeated co-occurrence" (p 59). Perhaps surprisingly, however, the results in Section 6 show that term repetition alone, independent of other discourse cues, can be a very useful indicator of subtopic structure. This may be less true in the case of narrative texts, which tend to use more variation in the way concepts are expressed, and so may require that thesaural relations be used as well, as in (Kozima, 1993).

It should be noted that other researchers have experimented with the display of patterns of cohesion cues other than lexical cohesion as tools for analyzing discourse structure. Grimes (1975)(Ch. 6) introduces "span charts" to show the interaction of various thematic devices such as character identification, setting, and tense. Stoddard (1991) creates "cohesion maps" by assigning to each word a location on a two-dimensional grid corresponding to the word's position in the text.

To summarize, many discourse analysis tasks require a fine-grained, hierarchical model, and consequently require many kinds of discourse cues for segmentation in practice. TextTiling attempts a coarser grained analysis and so gets away with using a simpler feature set. Additionally, if we think of subtopic segmentation in terms of detection of *shift* from one discussion to the next, we can simplify the task to be one of detecting where the use of one set of terms ends and another set begins. Figure 2 illustrates that lexical distribution information can be used to discover such subtopic shifts.

The next subsections describe three different strategies for detecting subtopic shift. The first is based on the observations of this subsection, that subtopics can be viewed as "clumps" of vocabulary, and the problem of segmentation is one of detecting these clumps. The following two subsections describe alternative techniques, derived by recasting other researchers' algorithms into a more appropriate framework for the TextTiling task.

4.1 Comparing Adjacent Blocks of Text

In the *block comparison* algorithm, adjacent pairs of text blocks are compared for overall lexical similarity. The TextTiling algorithm requires that a score, called the *lexical score*, be computed for every sentence, or more precisely, for the gap between every pair of sentences (since this is where paragraph breaks take place).

The sketch in Figure 3(a) illustrates the scores computed for the block comparison algorithm. In this figure is shown a sequence of eight hypothetical sentences, their contents represented as columns of letters, where each letter represents a term or word. The sentences are grouped into blocks of size k, where in this illustration $k = 2$. The more words the blocks have in common, the higher the lexical score at the gap between them. If a low lexical score is preceded by and followed by high lexical scores, this is assumed to indicate a shift in vocabulary corresponding to a subtopic change.

The blocks act as moving windows over the text. Several sentences can be contained within a block, but the blocks shift by only one sentence at a time. Thus if there are k sentences within a block, each sentence occurs in $k * 2$ score computations (except for sentences at the extreme ends of the text).

The current version of the block algorithm computes scores in a very simple manner, as the inner product of two vectors, where a vector contains the number of times each lexical item occurs in its corresponding block. The inner product is normalized to make the score fall between 0 and 1 inclusive.

Figure 3(a) shows the computation of the scores at the gaps between sentences 2 and 3, between 4 and 5, and between 6 and 7. The scores shown are simple, unnormalized

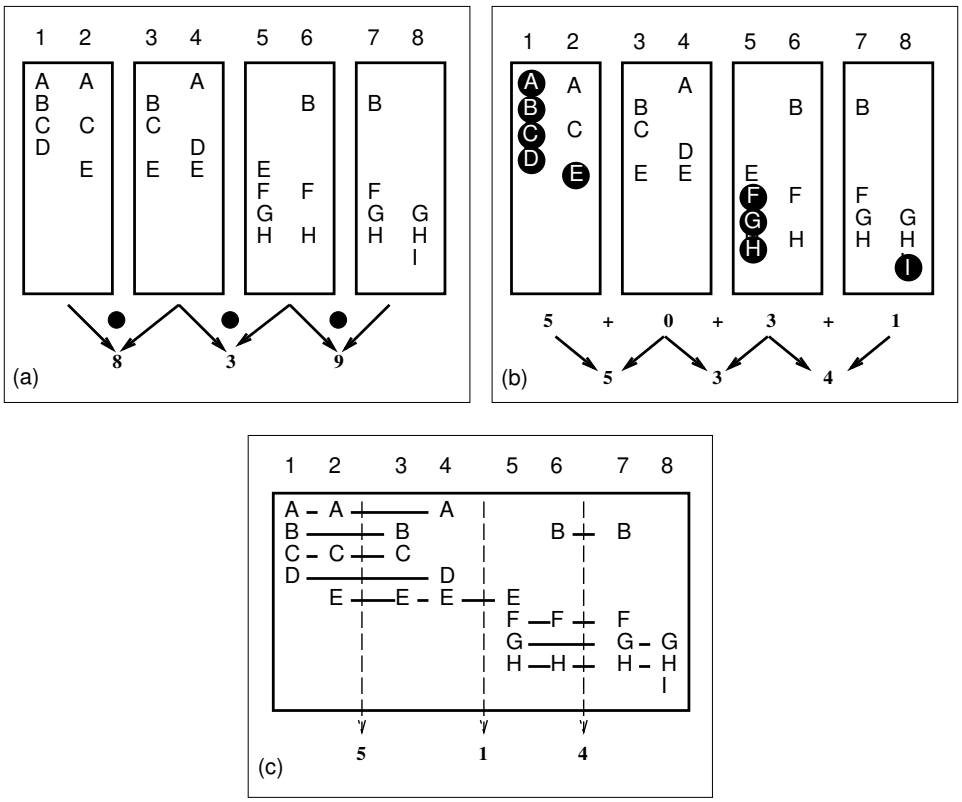

Figure 3
Illustration of three ways to compute the *lexical score* at gaps between sentences. Numbers indicate a numbered sequence of sentences, columns of letters signify the terms in the given sentence. (a) Blocks – dot product of vectors of word counts in the block on the left and the block on the right. (b) Vocabulary introduction – the number of words that occur for the first time within the interval centered at the sentence gap. (c) Chains – the number of active chains, or terms that repeat within threshold sentences and span the sentence gap.

inner products of the frequencies of the terms in the blocks. For example the gap between sentences 2 and 3 gets assigned a score of 8 computed as $2 * 1$ (for A) $+1 * 1$ (for B) $+2 * 1$ (for C) $+1 * 1$ (for D) $+1 * 2$ (for E). Results for this approach are reported in Section 6.

After these scores are computed, the blocks are shifted by one sentence (sentences 1 and 8 need to be handled as boundary conditions). So, for example, in addition to comparing sentences 3 and 4 against sentences 5 and 6, the algorithm compares sentences 4 and 5 against sentences 6 and 7.

An earlier version of the algorithm presented in (Hearst, 1993; Hearst and Plaunt, 1993) weighted terms according to *tf.idf* weights from Information Retrieval (Salton, 1989). This weighting function computes, for each word, the number of times it occurs in the document *tf*, times the inverse of the number of documents that the term occurs in a large collection *idf*, or as in this case, with some normalizing constants. The idea is that terms that occur commonly throughout a collection are not necessary good indicators of relevance to a query because they are so common, and so their importance is downweighted. Hearst (1993) posited that this argument should also apply to determining which words best distinguish one subtopic from another. However, it seems to be the case that the estimates of importance that *tf.idf* makes are not accurate enough within the scope of comparing adjacent pieces of text to justify using this measure, and

the results seem more robust weighting the words according to their frequency within the block alone.

4.2 Vocabulary Introductions

Another recent analytic technique that makes use of lexical information is described in Youmans (1991), which introduces a variant on type/token curves, called the Vocabulary-Management Profile. Type/token curves are simply plots of the number of unique words against the number of words in a text, starting with the first word and proceeding through the last. Youmans modifies this algorithm to keep track of how many first-time uses of words occur at the midpoint of every 35-word window in a text. Youmans' goal is to study the distribution of vocabulary in discourse rather than to segment it along topical lines, but upon examining many English narratives, essays, and transcripts he notices that sharp upturns after deep valleys in the curve "correlate closely to constituent boundaries and information flow."

Youmans' analysis of the graphs is descriptive in nature, mainly attempting to identify the cause of each peak or valley in terms of a principle of narrative structure, and is done at a very fined-grained level. He discusses one text in detail, describing changes at the single-word level, and focusing on within-paragraph and within-sentence events. Examples of events are changes in characters, occurrences of dialogue, and descriptions of places, each of which range in length from one clause to a few sentences. He also finds that paragraph boundaries are not always predicted – sometimes the onset of a new paragraph is signalled by the occurrence of a valley in the graph, but often paragraph onset is not signalled until one or two sentences beyond onset.[6]

One of Youmans' main foci is an attempt to cast the resulting peaks in terms of coordination and subordination relations. However, in the discussion he notes that this does not seem like an appropriate use of the graphs. No systematic evaluation of the algorithm is presented, nor is there any discussion of how one might automatically determine the significance of the peaks and valleys.

Nomoto and Nitta (1994) attempt to use Youmans' algorithm for distinguishing entire articles from one another when they are concatenated into a single file. They find that it "fails to detect any significant pattern in the corpus." I recast Youmans' algorithm into the TextTiling framework, renaming it the *vocabulary introduction* method. Figure 3(b) illustrates. The text is analyzed, and the positions at which terms are first introduced are recorded (shown in black circles in the figure). A moving window is used again, as in the blocks algorithm, and this window corresponds to Youmans' interval. The number of new terms that occur on either side of the midpoint, or the sentence gap of interest, are added together and plotted against sentence gap number.

This approach differs from that of Youmans (1991) and Nomoto and Nitta (1994) in two main ways. First, Nomoto and Nitta (1994) use too large of an interval – 300 words – because this is approximately the average size needed for their implementation of the blocks version of TextTiling. Large paragraph-sized intervals for measuring introduction of new words seems unlikely to be useful since every paragraph of of a given length should have approximately the same number of new words, although those at the beginning of a subtopic segment will probably have slightly more. Instead, I use interval lengths of size 40, closer to Youmans' suggestion of 35.

Second, the granularity at which Youmans takes measurements is too fine, since he plots the score at every word. Sampling this frequently yields a very spiky plot from

6 This might be explained in part by (Stark, 1988) who shows that readers disagree measurably about where to place paragraph boundaries when presented with texts with those boundaries removed.

which it is quite difficult to draw conclusions at a paragraph-sized granularity. By contrast I plot the score at every sentence gap, thus eliminating the wide variation that is seen when measuring after each word. Results for this approach are reported in Section 6.

4.3 Lexical Chains

Morris and Hirst's pioneering work on computing discourse structure from lexical relations (Morris and Hirst, 1991; Morris, 1988) is a precursor to the work reported on here. Influenced by Halliday and Hasan's (1976) theory of lexical coherence, Morris developed an algorithm that finds chains of related terms via a comprehensive thesaurus (Roget's Fourth Edition).[7] For example, the words *residential* and *apartment* both index the same thesaural category and can thus be considered to be in a coherence relation with one another. The chains are used to structure texts according to the attentional/intentional theory of discourse structure (Grosz and Sidner, 1986) discussed above. The extent of the lexical chains is assumed to correspond to the extent of a segment. The algorithm also incorporates the notion of "chain returns" – repetition of terms after a long hiatus – to complete an intention that spans over a digression. The boundaries of the segments corresponds to the sentences that the first and last words of the chain occur in.

Since the Morris and Hirst (1991) algorithm attempts to discover attentional / intentional structure, its goals are different than those of TextTiling. Specifically, the discourse structure it attempts to discover is hierarchical and more fine-grained than that discussed here. Morris (1988) provides five short example texts for which she has determined the intentional structure, and states that the lexical chains generated by her algorithm provide a good indication of the segment boundaries that Grosz and Sidner's theory assumes. In (Morris, 1988; Morris and Hirst, 1991), tables are presented showing the sentences spanned by the lexical chains and by the corresponding segments of the attentional/intentional structure (derived by hand), but no formal evaluation is performed.

This algorithm is not directly applicable for TextTiling for several reasons. First, many words are ambiguous and fall into more than one thesaurus class. This is not stated as a concern in Morris's work, perhaps because the texts were short, and presumably if a word were ambiguous the correct thesaurus class would nevertheless be chosen because the chained-to words would share only the correct thesaurus class. However, my experimentation with an implemented version of Morris' algorithm that made use of Roget's 1911 thesaurus (which is admittedly less structured than the thesaurus used by Morris), when run on longer texts, found ambiguous links to be a common occurrence and detrimental to the algorithm. A thesaurus-based disambiguation algorithm (Yarowsky, 1992) may help alleviate this problem (this option is revisited in Section 7), but another solution is to move away from thesaurus classes and use simple word co-occurrence instead, since within a given text a word is usually used with only one sense (Gale, Church, and Yarowsky, 1992b). The potential downside of this approach is that many useful links may be missed.

Another limitation of the Morris algorithm is that it does not take advantage of, or discuss how to account for, the tendency for multiple simultaneous chains to occur over the same intention (each chain corresponds to one intention). Related to this is the fact that chains tend to overlap one another in long texts, as can be seen in Figure 2.

These two types of difficulties can be circumvented by recasting the Morris algorithm to take advantage of the observations of Subsection 4. Three changes are made to

7 The algorithm was executed by hand since the thesaurus is not generally available online. Current extensions to this work make use of WordNet (Miller et al., 1990).

the algorithm. First, no thesaurus classes are used; only term repetition on morphological variants of the same word. Second, multiple chains are allowed to span an intention, and third, chains at all levels of intentions are analyzed simultaneously. Instead of deciding which chain is the applicable one for a given intention, it measures how many chains at all levels are active at each sentence gap. This approach is illustrated in Figure 3(c). A lexical chain for term t is considered active across a sentence gap if instances of t occur within some distance threshold of one another. In the figure, all three instances of the word A occur within the distance threshold. The third B, however, follows too far after the second B to continue the chain. The score for the gap between 2 and 3 is simply the number of active chains that span this gap. Boundaries are determined as specified in Section 5. This variation of the TextTiling algorithm is explored and evaluated in (Hearst, 1994b).

4.4 Vector Space Similarity Comparisons

As mentioned in Section 2, Salton and Allan (1993) report work in the automatic detection of hypertext links and theme generation from large documents, focusing primarily on encyclopedia text. They describe the application of similarity comparisons between articles, sections, and paragraphs within an encyclopedia, both for creating links among related passages, and for better facilitating retrieval of articles in response to user queries. Their approach finds similarities among the paragraphs of large documents using normalized *tf.idf* term weighting, scoring text segments according to a normalized inner product of vectors of these weights (this algorithm is called the *vector space* model (Salton, 1989)).

However, Salton and Allan (1993) do not try to determine the extents of passages within articles or sections. Instead, all paragraphs, sections, and articles are assigned pairwise similarity scores, and links are drawn between those with the highest scores, independent of their position within the text. This distinction is important because the difficulty in subtopic segmentation lies in detecting the subtle differences between adjacent text blocks. A method that finds blocks with the topmost similarity to one another can succeed at finding the equivalent of the *center* of a subtopic extent, but does not distinguish where one subtopic ends and the next begins.

If the algorithm of Salton and Allan (1993) were transformed so that adjacent text units were compared and a method for determining where the similarity scores are low is used, then it would resemble the blocks algorithm with *tf.idf* weighting, but without the use of overlapping text windows. However, as a consequence of the fact that the vector space method is better at distinguishing similarities than differences, similarity scores alone are probably less effective at finding the transition points between subtopic discussions than sequences of similarity scores, using moving windows of text, in the manner described above.

Salton et al. (1996) attempt to address a version of the subtopic segmentation problem by extending the algorithm to finding "text pieces exhibiting internal consistency that can be distinguished from the remainder of the surrounding text." As one part of this goal, they seek what is called the *text segment*, which is defined as "a contiguous piece of text that is linked internally, but largely disconnected from the adjacent text. Typically, a segment might consist of introductory material, or cover the exposition and development of the text, or contain conclusions and results." Thus they attempt only to find those segments of text that are strongly different than the surrounding text (and thus do not address the subtopic detection task). They do this by comparing similarity between a paragraph and its four closest paragraph neighbors to the left and the right. If a similarity score between a pair of paragraphs does not exceed a threshold, then the link between that pair is removed. If a disconnected sequence of paragraphs is found, that sequence

is considered a text segment. This algorithm is not evaluated.

4.5 Other Related Approaches

Kozima (1993) describes an algorithm for the detection of text segments , which are defined as "a sequence of clauses or sentences that display local coherence" in narrative text. Kozima (1993) presents a very elaborate algorithm for computing the lexical cohesiveness of a window of words, using spreading activation in a semantic network created from an English dictionary. The cohesion score is plotted against words and smoothed, and boundaries are considered to fall at the lowest scoring words. This complex computation may be necessary, as opposed to simple term repetition, when working with narrative texts, but no comparison of methods is done. The algorithm's results are shown on one text, but are not evaluated formally.

Reynar (1994) describes an algorithm similar to that of Hearst (1993), Hearst and Plaunt (1993) with a difference in the way in which the size of the blocks of adjacent regions are chosen. A greedy algorithm is used: first, the algorithm begins with no boundaries, then a boundary b (between two sentences) is chosen which maximizes the lexical score resulting from comparing the block on the left whose extent ranges from b to the closest existing boundary on the left, and similarly for the right. This process is repeated until a pre-specified number of boundaries have been chosen. This seems problematic since the initial comparisons are between very large text segments: the first boundary is chosen by comparing the entire text to the right and left of the initial position. The algorithm is evaluated only in terms of how well it distinguishes entire articles from one another when concatenated into one file. The precision/recall tradeoffs varied widely: on 660 *Wall Street Journal* articles, if the algorithm is allowed to be off by up to three sentences, it achieves precision of $.80$ with recall of $.30$, and precision of $.30$ with recall of $.92$.

5. The TextTiling Algorithm

The TextTiling algorithm for discovering subtopic structure using term repetition has three main parts:

1 Tokenization
2 Lexical Score Determination
3 Boundary Identification

Each is discussed in turn below. The methods for lexical score determination were outlined in Section 4, but more detail is presented here.

5.1 Tokenization

Tokenization refers to the division of the input text into individual lexical units, and is sensitive to the format of the input text. For example, if the document has markup information, the header and other auxiliary information is skipped until the body of the text is located. Tokens that appear in the body of the text are converted to all lower-case characters and checked against a "stop list" of closed-classed and other high frequency words.[8] If the token is a stop word then it is not passed on to the next step. Otherwise, the token is reduced to its root by a morphological analysis function based on that of Karttunen, Koskenniemi, and Kaplan (1987), converting regularly and irregularly inflected nouns and verbs to their roots.

8 "Stop list" is a term commonly used in Information Retrieval (Salton, 1989). In this case, the list consists of 898 words, developed in a somewhat ad hoc manner.

The text is subdivided into pseudosentences of a pre-defined size w (a parameter of the algorithm) rather than using "real" syntactically-determined sentences. This is done to allow for comparison between equal-sized units, since the number of shared terms between two long sentences and between a long and a short sentence would probably yield incomparable scores (and sentences are too short to expect normalization to really accommodate for the differences). For the purposes of the rest of the discussion these groupings of tokens will be referred to as *token-sequences*. The morphologically-analyzed token is stored in a table along with a record of the token-sequence number it occurred in, and the number of times it appeared in the token-sequence. A record is also kept of the locations of the paragraph breaks within the text. Stop words contribute to the computation of the size of the token-sequence, but not to the computation of the similarity between blocks of text.

5.2 Determining Scores

As mentioned above, two methods for determining the score to be assigned at each token sequence gap are explored here. The first, block comparison, compares adjacent blocks of text to see how similar they are according to how many words the adjacent blocks have in common. The second, the vocabulary introduction method, assigns a score to a token-sequence gap based on how many new words were seen in the interval in which it is the midpoint.

5.2.1 Blocks. In the block comparison algorithm, adjacent pairs of blocks of token-sequences are compared for overall lexical similarity. The *block size*, labeled k, is the number of token-sequences that are grouped together into a block to be compared against an adjacent group of token-sequences. This value is meant to approximate the average paragraph length. Actual paragraphs are not used because their lengths can be highly irregular, leading to unbalanced comparisons, but perhaps with a clever normalizing scheme, "real" paragraphs could be used (analogous to the substitution of token-sequences for real sentences).

Similarity values are computed for every token-sequence gap number; that is, a score is assigned to token-sequence gap i corresponding to how similar the token-sequences from token-sequence $i-k$ through i are to the token-sequences from $i+1$ to $i+k+1$. Note that this moving window approach means that each token-sequence appears in $k*2$ similarity computations.

The lexical score for the similarity between blocks is calculated by a normalized inner product: given two text blocks b_1 and b_2, each with k token-sequences, where $b_1 = \{tokens_{i-k}, ..., tokens_i\}$ and $b_2 = \{tokens_{i+1}, ..., tokens_{i+k+1}\}$,

$$score(i) = \frac{\sum_t w_{t,b_1} w_{t,b_2}}{\sqrt{\sum_t w_{t,b_1}^2 \sum_t w_{t,b_2}^2}}$$

where t ranges over all the terms that have been registered during the tokenization step (thus excluding stop words), and $w_{t,b}$ is the weight assigned to term t in block b. As mentioned in Section 4, in this version of the algorithm, the weights on the terms are simply their frequency within the block. This formula yields a score between 0 and 1, inclusive.

These scores can be plotted, token-sequence number against similarity score. However, since similarity is measured between blocks b_1 and b_2, the score's x-axis coordinate falls between token-sequences i and $i+1$. Rather than plotting a token-sequence number on the x-axis, the token-sequence *gap* number i is plotted instead.

5.2.2 Vocabulary Introduction. The lexical score assigned in the vocabulary introduction version of scoring is the ratio of new words in an interval divided by the length of that interval. Tokenization is as described above, eliminating stop words and performing morphological analysis. A score is then assigned to a token-sequence gap as follows: the number of never-yet seen words in the token-sequence to the left of the gap is added to the number of never-yet seen words in the token-sequence to the right, and this number is divided by the total number of tokens in the two token-sequences, or $w * 2$. Since in these experiments w is set to 20, this yields an interval length of 40, which is close to the parameter 35 suggested as most useful in (Youmans, 1991). As in the block version of the algorithm, the score is plotted at the token-sequence gap, and scores can range from 0 to 1, inclusive.

The lexical score is computed as follows. For each token-sequence gap i, create a text interval b of length $w * 2$ (where w is the length of the token-sequences) centered around i, and let b be subdivided into two equal-length parts, $b1$ and $b2$, where $b_1 = \{tokens_{i-w}, ..., tokens_i\}$ and $b_2 = \{tokens_{i+1}, ..., tokens_{i+w+1}\}$. Then,

$$score(i) = \frac{NumNewTerms(b_1) + NumNewTerms(b_2)}{w * 2}$$

where $NumNewTerms(b)$ returns the number of terms in interval b seen for the first time in the text.

5.3 Boundary Identification

Boundary identification is done identically for all lexical scoring methods, and assigns a *depth score*, the depth of the valley (if one occurs), to each token-sequence gap. The depth score corresponds to how strongly the cues for a subtopic changed on both sides of a given token-sequence gap and is based on the distance from the peaks on both sides of the valley to that valley. Figure 4 illustrates. In Figure 4(a), the depth score at gap $a2$ is $(y_{a1} - y_{a2}) + (y_{a3} - y_{a2})$. Relatively "deeper" valleys receive higher scores than shallower ones. More formally, for a given token-sequence gap i, the program records the lexical score of the token-sequence gap l to the left of i until the score for $l - 1$ is smaller than the score for l (meaning the top of the peak was found at l). Similarly, for token sequences to the right of i, the program monitors the score of token-sequence r until the score for $r + 1$ is less than that of r. Finally, $score(r) - score(i)$ is added to $score(l) - score(i)$, and the result is the depth score at i.

A potential problem with this scoring method is illustrated in Figure 4(b). Here we see a small valley at gap $b4$ that can be said to "interrupt" the score for $b2$. As one safeguard, the algorithm uses smoothing (described below) to help eliminate small pertubations of the kind seen at $b4$. Additionally, because the distance between y_{b3} and y_{b4} is small in these kinds of cases, this gap is less likely to be marked as a boundary than gaps like $b2$, which have large peak distances both to the left and the right. This example illustrates the need to take into account the length of both sides of the valley, since a valley that has high peaks on both sides indicates that not only has the vocabulary on the left decreased in score, but the vocabulary on the right has increasing score, thus signalling a strong subtopic change.

Figure 4(c) shows another potentially problematic case, in which two strong peaks flank a long, flat valley. The question becomes which of gaps $c2$, $c3$, or both, should be assigned a boundary. Such "plateaus" occur when vocabulary changes very gradually, and reflect a poor fit of the corresponding portion of the document to the model assumed by TextTiling. When the plateau occurs over a longer stretch, usually it is reasonable to choose both bordering gaps as boundaries. However, when such a plateau occurs over a very short stretch of text, the algorithm is forced to make a somewhat arbitrary choice.

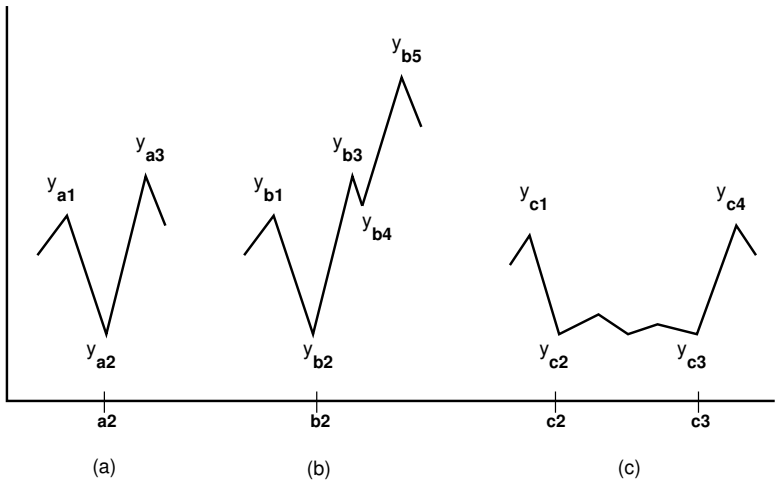

Figure 4
A sketch illustrating the computation of depth scores in three different situations. The x-axis indicates token sequence gap number and the y-axis indicates lexical score.

Choices like these are cases in which the algorithm should probably make use of additional information, such as more localized lexical distribution information, or perhaps more conventional discourse cues.

Note that the depth scores are based only on relative score information, ignoring absolute values. The justification for this is twofold. First, it helps make decisions in the cases in which a gap's lexical score falls into the middle of the lexical score range, but is flanked by tall peaks on either side, and this situation happens commonly enough to be important. Second, using relative rather than absolute scores helps avoid problems associated with situations like that of Figure 4(c), in which all gaps between $c2$ and $c3$ would be considered boundaries if only absolute scores were taken into account.

The depth scores are sorted and used to determine segment boundaries. The larger the score, the more likely the boundary occurs at that location, modulo adjustments as necessary to place the boundaries at orthographically marked paragraphs (if available). A proviso check is made to prevent assignment of very close adjacent segment boundaries. Currently at least three intervening token-sequences are required between boundaries. This helps control for the fact that many texts have spurious header information and single-sentence paragraphs.

An alternative to this method of computing depth scores is to use the slope of the valley's sides, or the "sharpness" of the vocabulary change. However, because deeper valleys with smaller slopes indicate larger, although more gradual, shifts in vocabulary usage than shallower valleys with larger slopes, they are preferable for detecting subtopic boundaries. Furthermore, steep slopes can sometime indicate a spurious change associated with a very short digression. The depth score is more robust for the purposes of subtopic boundary detection.

5.4 Smoothing the Plot

As mentioned above, the plot is smoothed to remove small dips, using average smoothing with a width of size s, as follows:

for each token-sequence gap g and a small even number s
find the scores of the $s/2$ gaps to the left of g
find the scores of the $s/2$ gaps to the right of g

find the score at g
take the average of these scores and assign it to g
repeat this procedure n times

The choice of smoothing function is somewhat arbitrary; other low-pass filters could be used instead.

5.5 Determining the Number of Boundaries

The algorithm must determine how many segments to assign to a document, since every paragraph is a potential segment boundary. Any attempt to make an absolute cutoff, even one normalized for the length of the document, is problematic since there should be some relationship between the structure and style of the text and the number of segments assigned to it. As discussed above, a cutoff based on a particular valley depth is similarly problematic.

Instead, I suggest making the cutoff a function of the characteristics of the depth scores for a given document, using the average $\bar{s}$ and standard deviation σ of their scores (thus assuming that the scores are normally distributed). One version of this function entails drawing a boundary only if the depth score exceeds $\bar{s} - \sigma$ (the liberal measure, LC). This function can be varied to achieve correspondingly varying precision/recall tradeoffs. A higher precision but lower recall can be found by setting the limit to be depth scores exceeding $\bar{s} - \sigma/2$ (the conservative measure, HC) instead of $\bar{s} - \sigma$.

6. Evaluation

There are several ways to evaluate a segmentation algorithm, including comparing its segmentation against that of human judges, comparing its segmentation against author-specified orthographic information, and comparing its segmentation against other automated segmentation strategies in terms of how they effect the outcome of some computational task. This section presents comparisons of the results of the algorithm against human judgments and against article boundaries. It is possible to compare against author-specified markups, but unfortunately, as discussed above, authors usually do not specify the kind of subtopic information desired. As mentioned above, Hearst (1995), Hearst and Plaunt (1993) show how to use TextTiles in information retrieval tasks, although this work does not show whether or not the results of these algorithms produce better performance than the results of some other segmentation strategy would.

6.1 Reader Judgments

There is a growing concern surrounding issues of inter-coder reliability when using human judgments to evaluate discourse processing algorithms (Carletta, 1996; Condon and Cech, 1995). Proposals have recently been made for protocols for the collection of human discourse segmentation data (Nakatani et al., 1995) and for how to evaluate the validity of judgments so obtained (Carletta, 1996; Isard and Carletta, 1995; Rosé, 1995; Passonneau and Litman, 1993; Litman and Passonneau, 1995). Recently, Hirschberg and Nakatani (1996) have reported promising results for obtaining higher inter-judge agreement using their collection protocols.

For the evaluation of the TextTiling algorithms, judgments were obtained from seven readers for each of twelve magazine articles which satisfied the length criteria (between 1800 and 2500 words)[9] and which contain little structural demarcation. The judges were

9 One longer text of 2932 words was used since reader judgments had been obtained for it from an earlier experiment. Judges were technical researchers. Two texts had three or four short headers which were

asked simply to mark the paragraph boundaries at which the topic changed; they were not given more explicit instructions about the granularity of the segmentation.[10]

Figure 5 shows the boundaries marked by seven judges on the *Stargazers* text. This format helps illustrate the general trends in the judges' assessments, and also helps show where and how often they disagree. For instance, all but one judge marked a boundary between paragraphs 2 and 3. The dissenting judge did mark a boundary after 3, as did two of the concurring judges. The next three major boundaries occur after paragraphs 5, 9, 12, and 13. There is some contention in the later paragraphs; three readers marked both 16 and 18, two marked 18 alone, and two marked 17 alone. The outline in the Introduction gives an idea of what each segment is about.

Passonneau and Litman (1993) discuss at length considerations about evaluating segmentation algorithms according to reader judgment information. As Figure 5 shows, agreement among judges is imperfect, but trends can be discerned. In the data of Passonneau and Litman (1993), if 4 or more out of 7 judges mark a boundary, the segmentation is found to be significant using a variation of the Q-test (Cochran, 1950). However, in later work (Litman and Passonneau, 1995), 3 out of 7 judges marking a boundary was considered sufficient to classify that point as a "major" boundary.

Carletta (1996) and Rosé (1995) point out the importance of taking into account the *expected chance* agreement among judges when computing whether or not judges agree significantly. They suggest using the *kappa* coefficient (κ) for this purpose. According to Carletta (1996), κ measures pairwise agreement among a set of coders making category judgments, correcting for expected chance agreement as follows:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where P(A) is the proportion of times that the coders agree and P(E) is the proportion of times that they would be expected to agree by chance. The coefficient can be computed by making pairwise comparisons against an expert or by comparing to a group decision. Carletta (1996) also states that in the behavioral sciences, $\kappa > .8$ signals good replicability, and $.67 < \kappa < .8$ allows tentative conclusions to be drawn. The kappa coefficients found in Isard and Carletta (1995) ranged from $.43$ to $.68$ for four coders placing transaction boundaries, and those found in (Rosé, 1995) ranged from $.65$ to $.90$ for four coders segmenting sentences. Carletta cautions, however, that "... coding discourse and dialogue phenomena, and especially coding segment boundaries, may be inherently more difficult than many previous types of content analysis (for instance, dividing newspaper articles based on subject matter)" and so implies that the levels of agreement needed to indicate good reliability for TextTiling may be justified in being lower.

For my test texts the judges placed boundaries on average 39.1% of the time, and nonboundaries 60.9%. Thus the expected chance agreement $P(E)$ is $.524$ (since $P(Boundary) = .391$ and $P(NonBoundary) = .609, (.391^2 + .609^2) = .524$). To compute κ, each judge's decision was compared to the group decision, where a paragraph gap was considered a "true" boundary if at least three out of seven judges placed a boundary

removed for consistency. One text that was used in (Hearst, 1994b) is not used here because inconsistencies were found in the paragraph break locations.

10 Specifically, the instructions were in written form and ran as follows: "You will receive three texts. Mark where the topics seem to change – draw a line between the paragraphs, where any blank line can be considered a paragraph boundary. It's recommended that you read quickly; no need to understand all the nuances. However, you are allowed to go back and look over parts that you've already looked at and change your markings if desired. If on occasion you can't decide between two places, definitely pick one but indicate that you thought the other one was just as appropriate." On the rare occasions in which the subject picked a secondary boundary, only the primary one was retained for evaluation.

mark there, as in Litman and Passonneau (1995).[11] The remaining gaps are considered nonboundaries. The average κ for these texts was $.647$. This score is at the low end of the stated acceptability range but is comparable with those of other inter-reliability results (with fewer judges) found in discourse segmentation experiments.

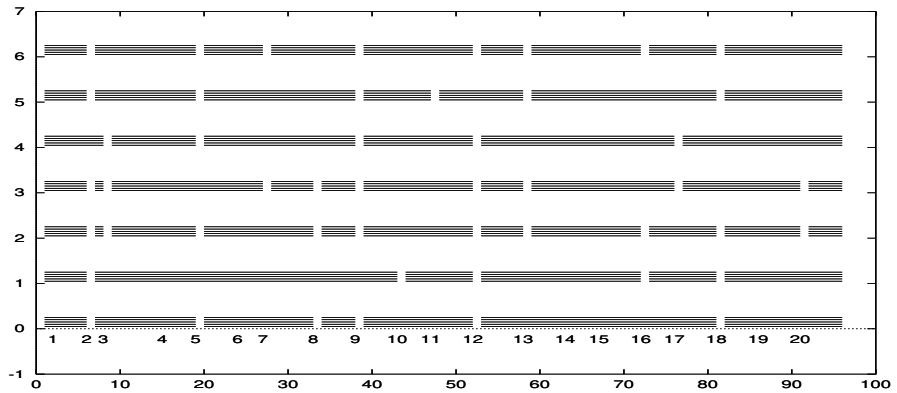

Figure 5
Judgments of seven readers on the *Stargazer* text. Internal numbers indicate location of gaps between paragraphs; x-axis indicates token-sequence gap number, y-axis indicates judge number, a break in a horizontal line indicates a judge-specified segment break.

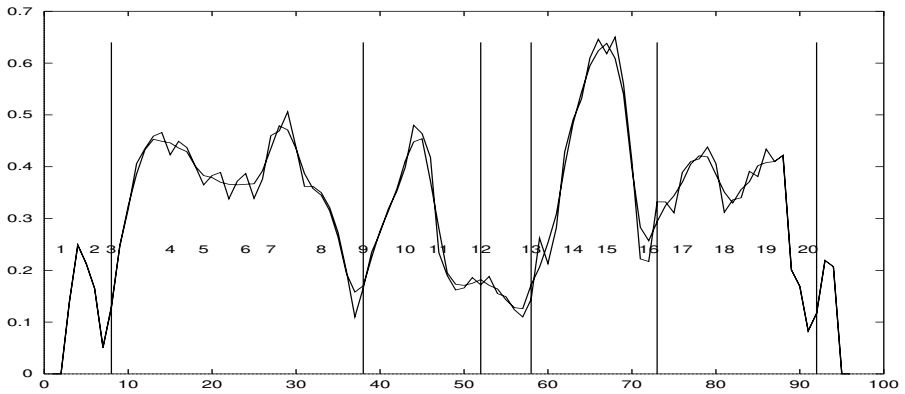

Figure 6
Results of the block similarity algorithm on the *Stargazer* text with k set to 10 and the loose boundary cutoff limit. Both the smoothed and unsmoothed plot are shown. Internal numbers indicate paragraph numbers, x-axis indicates token-sequence gap number, y-axis indicates similarity between blocks centered at the corresponding token-sequence gap. Vertical lines indicate boundaries chosen by the algorithm; for example, the leftmost vertical line represents a boundary after paragraph 3. Note how these align with the boundary gaps of Figure 5 above.

6.2 Parameter Settings

An unfortunate aspect of the algorithm in its current form is that it requires the setting of several inter-dependent parameters, the most important of which are the size of the text unit that is compared, and the number of words in a token-sequence (which controls the number of times a term appears in a window as well as the number of data points

11 Paragraphs of three or fewer sentences were combined with their neighbor if that neighbor was deemed to follow at a "major" boundary, as in paragraphs 2 and 3 of the *Stargazers* text.

that are sampled). The method, width, and number of rounds of smoothing must also be chosen. Usually only modest amounts of smoothing can be allowed, since more dramatic smoothing tends to obscure the point at which the subtopic transition takes place. Finally, the method for determining how many boundaries to assign must be specified. The three are inter-related: for example, using a larger text window requires less smoothing and fewer boundaries will be found, yielded a coarser-grained segmentation.

Initial testing was done on the texts evaluated with several different sets of parameter settings and a default configuration that seems to cover many different text types was chosen. The defaults set $w = 20, k = 10, n = 1, s = 2$, for token-sequence size, block size, number of rounds of smoothing, and smoothing width, respectively. The evaluation presented here show the results for different setting types to give a feeling for the space of results. Because the evaluation collection is very small, these results can be seen only as a suggestion; different settings may work better in different situations.

6.3 Results: Qualitative Analysis

Figure 6 shows a plot of the results of applying the block comparison algorithm to the *Stargazer* text with k set to 10. When the lowermost portion of a valley is not located at a paragraph gap, the judgment is moved to the nearest paragraph gap.[12] For the most part, the regions of strong similarity correspond to the regions of strong agreement among the readers. (The results for this text are among the stronger ones and appear in the last line of Table 2.) Note however, that the similarity information around paragraph 12 is weak. This paragraph briefly summarizes the contents of the previous three paragraphs; much of the terminology that occurred in all of them reappears in this one location (in the spirit of a Grosz and Sidner (1986) "pop" operation). Thus it displays low similarity both to itself and to its neighbors. This is an example of a breakdown caused by the assumptions about the subtopic structure.

Because of the depth score cutoff, not all valleys are chosen as boundaries. Although there is a dip around paragraph gaps 5 and 6, no boundary is marked there. From the summary of the text's contents in Section 1, we know that paragraphs 4-5 discuss the moon's chemical composition while 6-8 discuss how it got its shape; these two subtopic discussions are more similar to one another in content than they are to the subtopics on either side of them, thus accounting for the small change in similarity.

Five out of seven readers indicated a break between paragraphs 18 and 19. The algorithm registers a slight, but not significant valley at this point. Upon inspection it turns out that paragraph 19 really is a continuation of the discussion in 18, answering a question that is posed at the end of 18. However, paragraph 19 begins with an introductory phrase type that strongly signals a change in subtopic: "For the last two centuries, astronomers have studied ...".

The final paragraph is a summary of the entire text; the algorithm recognizes the change in terminology from the preceding paragraphs and marks a boundary; only two of the readers chose to differentiate the summary; for this reason the algorithm is judged to have made an error even though this sectioning decision is reasonable. This illustrates the inherent fallibility of testing against reader judgments, although in part this is because the judges were given loose constraints.

12 More specifically, if the closest paragraph location (first left, then right) has not been marked as a boundary, then mark it. Otherwise, look to the paragraph to the left. If that paragraph has not been marked and if it is at least $gap_limit = 3$ token-sequences away, then mark the paragraph to the left. If this fails, try the paragraph to the right in a similar way. If both fail, mark nothing.

Baseline		Tiling (V)						Tiling (B)						Judges		
		LC			HC			LC			HC					
$\bar{P}$	$\bar{R}$	κ	$\bar{P}$	$\bar{R}$	κ	$\bar{P}$	$\bar{R}$	κ	$\bar{P}$	$\bar{R}$	κ	$\bar{P}$	$\bar{R}$	κ	$\bar{P}$	$\bar{R}$
50	51	23	52	78	32	58	64	46	66	75	47	71	59	65	83	71

Table 1
Average κ, precision and recall scores for 12 test texts. Baseline shows the scores for an algorithm that assigns a boundary 39% of the time (the average overall), Tiling (V) indicates the vocabulary introduction version of computing lexical scores with token-sequence size $w = 20$, and Tiling (B) indicates the blocks version with token-sequence size $w = 20$ and block size $k = 10$. Both versions' results are shown at both the low cutoff (LC) and the high cutoff (HC) for terminating boundary assignment. Judges shows the average kappa, precision, and recall for all judges averaged over all texts.

6.4 Results: Quantitative Analysis

To assess the results of the algorithm quantitatively, I follow the advice of Gale, Church, and Yarowsky (1992a), and compare the algorithm against both upper and lower bounds. The upper bound in this case is the reader judgment data. The lower bound is a baseline algorithm that is a simple, reasonable approach to the problem that can be automated. A simple way to segment the texts is to place boundaries randomly in the document, constraining the number of boundaries to equal that of the average number of paragraph gaps per document assigned as boundaries by judges. In the test data, boundaries are placed in about 39% of the paragraph gaps. A program was written that places a boundary at each potential gap 39% of the time (using a random number generator), and run 10,000 times for each text, and the average of the scores of these runs was found. These scores appear in Table 1.

The algorithms are evaluated according to the proportion of "true" or majority boundaries they select out of the total selected (precision) and the proportion of "true" boundaries found out of the total possible (recall) (Salton, 1989). Precision also implies the number of extraneous boundaries (or false positives, or insertion errors), and recall implies the number of missed boundaries (or false negatives, or deletion errors).

Table 1 shows that both the blocks algorithm for lexical score assignment and the vocabulary introduction algorithm fall between the upper and lower bounds. The results are shown for making both a liberal (LC) and a conservative (HC) number of boundary assignments (see Subsection 5.5). As is to be expected, when more boundaries can be assigned, recall becomes higher at the expense of precision, and conversely, when boundary assignment is conservative, better precision is obtained at the expense of recall. This table also shows the average κ scores for the agreement between the algorithm and the judges. The scores for the blocks version of the algorithm are stronger than those of the vocabulary introduction version.

Table 2 shows results in more detail, varying some of the parameter settings. In order to allow for a more direct comparison, the precision for each version of the algorithm is shown at the recall level obtained by the judges on average. This is computed as follows for each version of the algorithm: the depth scores are examined in order of their strength. For each depth score, if it corresponds to a true boundary, the count of correct boundaries is incremented, otherwise the count of incorrect boundaries is incremented. Precision and recall are computed after each correct boundary encountered. When the recall equals that of the judges' average recall, the corresponding precision of the algorithm is returned. If the recall level exceeds that of the judges', then the value of the precision is estimated as a linear interpolation between the two precision scores whose recall

	NP	NB	JP	JR	κ	B7	B9	B10	B12	V10	V16	V20	V24
1	18	8	.809	.696	.56	.580	.580	.611	.524	.480	.500	.442	.442
2	30	10	.897	.714	.74	.877	—	—	.781	.505	.617	.633	—
3	21	9	.907	.778	.72	.875	.875	.875	.788	.583	.500	.778	.636
4	41	14	.892	.684	.68	.593	.577	.614	.790	.528	.558	.633	—
5	30	9	.716	.619	.72	.480	.687	.687	—	.478	.649	.500	.581
6	25	16	.932	.688	.52	—	—	—	—	.785	.785	—	—
7	39	8	.736	.732	.75	—	—	—	—	.422	.634	.402	.467
8	28	10	.793	.657	.63	1.0	1.0	.766	.766	.522	.464	.541	.450
9	27	11	.917	.649	.65	.682	.854	.781	.683	.460	.416	.704	.588
10	24	8	.743	.857	.67	.695	—	.707	.707	.478	—	—	.471
11	17	8	.812	.768	.61	—	.605	.544	—	.380	.458	.591	.662
12	21	9	.839	.651	.58	.673	.673	.745	.745	.500	.604	.539	.455

Table 2
Precision for various parameter settings at the recall level obtained on average by the judges for twelve texts. NP: number of paragraphs; NB: number of boundaries according to judges' consensus; JP: judges' average precision; JR: judges' average recall; κ: kappa for the judges for each text; Bk: precision for the blocks algorithm with block size k and $w = 20$; Vw: precision for the vocabulary introduction algorithm with token sequence size w. Dashes occur in cases in which the algorithm does not produce a recall level equivalent to that of the judges' average.

scores most closely surround that of the judges' average recall. (This assumption of a linear interpolation is justified because in most, although not all, cases precision changes monotonically.) In some cases the algorithm does not produce a recall level as high as that found by the judges, since paragraphs with a non-positive depth score are not eligible for boundary assignment, and these cases are marked with a dash. Note that this evaluation does away with the need for LC and HC cutoff levels.

From Table 2 we can see that varying the parameter settings improves the scores for some texts while detracting from others. We can also see that the block algorithm for lexical score determination produces stronger results in most cases than the vocabulary introduction method, although the latter seems to do better on the cases where the blocks algorithm finds few boundaries (e.g., texts 6, 7, and 11). In almost all cases the algorithms are not as accurate as the judges, but the scores for the block version of the algorithm are very strong in many cases.

In looking at the results in more detail, one might wonder why the algorithm performs better on some texts than on others. Text 7, for example, scores especially poorly. This may be caused by the fact that this text has a markedly different style from the others. It is a chatty article (about how to survive office politics), and consists of a series of anecdotes about particular individuals. The article is interspersed throughout with spoken quotations, and these tend to throw the algorithm off because spoken statements usually contain different vocabulary than the surrounding prose. This phenomenon occurs in some of the other texts as well, but to a much lesser extent. It suggests a need for recognizing and accommodating very short digressions more effectively. Another interesting property of this text is that most of the subtopic switches occur when switching from one anecdote to another, and by inspection it appears that the best cues for these switches are pronouns which appear on the stop list and are discarded (for example, the anecdotes alternate between men and women's experiences, and correspondingly alternate between using *she* and *her* and using *he* and *him*). However, in most cases use of the stop list improves results.

It should also be noted that the texts used in this study were not chosen to have well-defined boundaries, and so pose a difficult test for the algorithm. Perhaps some tests

against texts with more obvious subtopic boundaries (for which the kappa coefficient for inter-judge agreement is larger) would be illuminating.

6.5 Detecting Breaks between Consecutive Documents

One way to evaluate the algorithm is in terms of how well it distinguishes entire articles from one another when they are concatenated into one file. Nomoto and Nitta (1994) implement the *tf.idf* version of TextTiling from (Hearst, 1993; Hearst and Plaunt, 1993) and evaluate it this way on Japanese newswire text[13]. Also, as discussed in Section 4, Reynar (1994) uses this form of evaluation on a greedy version of the blocks algorithm.

This task violates a major assumption of the TextTiling algorithm. TextTiling assumes that the similarity comparisons are done within the vocabulary patterns of one text, and so a *relatively* large shift in vocabulary indicates a change in subtopic. Because this evaluation method assumes that article boundary changes are more important than subtopic boundary changes, it penalizes the algorithm for marking very strong subtopic changes that occur within a very cohesive document before relatively weaker changes in vocabulary between similar articles. For example, for hypothetical articles $d1$, $d2$, and $d3$, assume $d1$ has very strong internal coherence indicators, $d2$ has relatively weak ones, and $d3$ is in the midrange. The interior subtopic transition scores for $d1$ can swamp out the score for the transition between $d2$ and $d3$.

Nevertheless, because others have used this evaluation method, one such evaluation is shown here as well. The evaluation set consisted of 44 articles from the Wall Street Journal from 1989. Consecutive articles were used, except any article that was fewer than 10 sentences was removed. The data consisted of 691 paragraphs, most of which contained between one and three sentences, some of which were very short, e.g., article bylines (thus making exact assignment of boundary locations more difficult). The text was not "clean": several articles consisted of a sequence of stories, several had tabular data, and one article was just a listing of interest rates.

The block version of TextTiling was run over this data using the default parameter settings. The depth scores were sorted and the number of assignments to article boundaries that were within three sentences of the correct location were recorded at several cutoff levels are shown in Table 3. B corresponds to the number of boundaries assigned, in sorted order (i.e., the first row shows the precision and recall after the first 10 boundaries are assigned), C corresponds to the number of correctly placed boundaries, P the precision, R the recall, and the asterisk shows the precision/recall breakeven point.

The higher scoring boundaries are almost always exact hits, but those farther down are more likely to be off by one to three sentences. Only one transition is missed entirely, and it occurs after a sequence of five isolated sentences and a byline (a weak boundary is marked preceding these isolated sentences). The high scoring boundaries that do not correspond to shifts between articles almost always correspond to strong subtopic shifts. One exception occurs in the article consisting only of interest rate listings. Another occurs in an article associating numerical information with names.

Overall the scores are much stronger than those reported in (Reynar, 1994), and are comparable to those of Nomoto and Nitta (1994) whose best precision/recall tradeoff on a collection of approximately 80 articles is approximately .50 precision and .81 recall. However, all three studies are done on different test collections and so comparisons are at best suggestive.

13 Instead of using fixed-sized blocks, Nomoto and Nitta (1994) take advantage of the fact that Japanese provides discourse markers that indicate multi-sentence units that participate in a topic/comment relationship, and find these motivated units can work slightly better.

B	C	P	R
10	8	.80	.19
20	16	.80	.37
30	22	.73	.51
40	27	.68	.63
43*	29	.67	.67
50	31	.62	.72
60	36	.60	.83
70	41	.59	.95

Table 3
Performance for block algorithm with default settings distinguishing between article boundaries in newspaper text consisting of 44 articles. B: number of boundaries chosen, C: number of correct boundaries, P: precision, R: recall.

7. Summary And Future Work

This article has described an algorithm that uses changes in patterns of lexical repetition as the cue for the segmentation of expository texts into multi-paragraph subtopic structure. It has also advocated the investigation and use of the multi-paragraph discourse unit, something that had not been explored in the computational literature until this work was introduced. The algorithms described here are fully implemented, and use term repetition alone, without requiring thesaural relations, knowledge bases, or inference mechanisms. Evaluation reveals acceptable performance when compared against human judgments of segmentation, although there is room for improvement.

TextTiles have already been integrated into a user interface in an information retrieval system (Hearst, 1995) and have been used successfully for segmenting Arabic newspaper texts (which have no paragraph breaks) for information retrieval (Hasnah, 1996). With the increase in importance of multimedia information, especially in the context of Digital Library projects, the need for segmentation and summarization of alternative media types is becoming increasingly important. For example, the algorithms described here should prove useful for topic-based segmentation of video transcripts (Christel et al., 1995). In a line of work we call Mixed-Media access (Chen et al., 1994), textual subtopic structure is being integrated with other media types, such as images and speech.

TextTiling has been used in innovative ways by other researchers. Karlgren (1996), in a study of the effects of stylistic variation in texts on information retrieval results, uses TextTiling as one of several ways of characterizing newspaper texts. Overall, he finds that relevant documents tend to be more complex than nonrelevant ones in terms of length, sentence structure, and other metrics. When examining documents of all lengths, he finds that relevant documents tend to have more TextTiles than nonrelevant ones (95% significant by a Mann Whitney test). As another example of an innovative application, van der Eijk (1994) suggests using TextTiles to align parallel multilingual text corpora according to the overlap in their subtopic structure for English, German, and French text. This work, along with that of Nomoto and Nitta (1994) on Japanese and Hasnah (1996) on Arabic, also provides evidence that TextTiling is applicable to a wide range of natural languages.

There are several ways that the algorithms could be modified in order to attempt to improve the results. One way is to use thesaural relations in addition to term repetition to make better estimates about the cohesiveness of the discussion. Earlier work

(Hearst, 1993) incorporated thesaural information into the algorithms, but later experiments found that this information degrades the performance. This could very well be due to problems with the thesaurus and assignment algorithm used. A simpler algorithm that just posits relations among terms that are a small distance apart according to WordNet (Miller et al., 1990) modeled after Morris and Hirst's heuristics, might work better. Therefore, the issue should not be considered closed, but rather as an area for future exploration, with this work as a baseline for comparison. The approach to similarity comparison suggested by Kozima (1993), while very expensive to compute, might also prove able to improve results. Other ways of computing semantic similarity, such as those of Schütze (1993) or Resnik (1995), may also prove useful. As a related point, experimentation should be done with variations in tokenization strategies, and it may be especially interesting to incorporate phrase or bigram information into the similarity computation.

The methods for computing lexical score also have the potential to be improved. Some possibilities are weighting terms according to their prior probabilities, weighting terms according to the distance from the location under scrutiny according to a Gaussian distribution, or treating the plot as a probabilistic time series and detecting the boundaries based on the likelihood of a transition from nontopic to topic. Another alternative is to devise a good normalization strategy that would allow for meaningful comparisons of "real" paragraphs, rather than regular sized windows of text.

The question arises as to how to extend the algorithm to capture hierarchical structure. One solution is to use the coarse subtopic structure to guide the more fine-grained methods. Another is to make several passes through the text, using the results of one round as the input, in terms of which blocks of text are compared, in the next round.

Finally, it may prove fruitful to use localized discourse cue information or other specialized processing around potential boundary locations to help better determine exactly where segmentation should take place. The use of discourse cues for detection of segment boundaries and other discourse purposes has been extensively researched, although predominantly on spoken text (see Hirschberg and Litman (1993) for a summary of six research groups' treatments of 64 cue words). It is possible that incorporation of such information may help improve the cases where the algorithm is off by one paragraph.

Acknowledgments. This article was enormously improved as a result of the careful comments of four anonymous reviewers, the editors of this special issue, and Christine Nakatani and Andreas Stölcke. Earlier writeups of this work benefited from the comments of Jan Pedersen, Per-Kristian Halvorsen, Ken Church, Bill Gale, David Yarowsky, Graeme Hirst, Jeff Siskind, Michael Braverman, Narciso Jaramillo, Dan Jurafsky, Mike Schiff, Dekai Wu, Penni Sibun, John Maxwell, Hinrich Schütze, and Christine Nakatani. I would like to thank Anne Fontaine for her interest and help in the early stages of this work, and Robert Wilensky for supporting this line of research as my thesis advisor. This work was sponsored in part by the Advanced Research Projects Agency under Grant No. MDA972-92-J-1029 with the Corporation for National Research Initiatives (CNRI), 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, and by the Xerox Palo Alto Research Center.

References

Brown, Gillian and George Yule. 1983. *Discourse Analysis*. Cambridge Textbooks in Linguistics Series. Cambridge University Press.

Callan, James P. 1994. Passage-level evidence in document retrieval. In *Proceedings of the 17th Annual International ACM/SIGIR Conference*, pages 302–310, Dublin, Ireland.

Carletta, Jean. 1996. Assessing agreement on classification tasks: The Kappa statistic. *Computational Linguistics*, 22(2):249–254.

Chafe, Wallace L. 1979. The flow of thought and the flow of language. In Talmy Givón, editor, *Syntax and Semantics: Discourse and Syntax*, volume 12. Academic Press, pages 159–182.

Chen, Francine, Marti A. Hearst, Julian Kupiec, Jan O. Pedersen, and Lynn Wilcox. 1994. Metadata in mixed-media access. *SIGMOD Record*, 23(4):64–71.

Chen, Francine R. and Margaret Withgott. 1992. The use of emphasis to automatically summarize a spoken discourse. In *ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 229–232.

Christel, M., T. Kanade, M. Mauldin, R. Reddy, M. Sirbu, S. Stevens, and H. Wactlar. 1995. Informedia digital video library. *Communications of the ACM*, 38(4):57–58, April.

Cochran, W. G. 1950. The comparison of percentages in matched samples. *Biometrika*, 37:256–266.

Condon, Sherri L. and Claude G. Cech. 1995. Problems for reliable discourse coding systems. In Johanna Moore and Marilyn Walker, editors, *Empirical Methods in Discourse: Interpretation & Generation, AAAI Technical Report SS-95-06*, Menlo Park, CA. AAAI Press.

Gale, William A., Kenneth W. Church, and David Yarowsky. 1992a. Estimating upper and lower bounds on the performance of word-sense disambiguation programs. In *Proceedings of the 30th Meeting of the Association for Computational Linguistics*, pages 249–256.

Gale, William A., Kenneth W. Church, and David Yarowsky. 1992b. One sense per discourse. In *Proceedings of the DARPA Speech and Natural Language Workshop*.

Girill, T. R. 1991. Information chunking as an interface design issue for full-text databases. In Martin Dillon, editor, *Interfaces for Information Retrieval and Online Systems*. Greenwood Press, New York, NY, pages 149–158.

Grimes, J. 1975. *The Thread of Discourse*. Mouton, The Hague.

Grosz, Barbara J. and Candace L. Sidner. 1986. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3):172–204.

Halliday, M. A. K. and R. Hasan. 1976. *Cohesion in English*. Longman, London.

Harman, Donna. 1991. How effective is suffixing? *Journal of the American Society for Information Science (JASIS)*, 42(1):7–15.

Harman, Donna. 1993. Overview of the first Text REtrieval Conference. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, pages 36–48, Pittsburgh, PA.

Hasnah, Ahmad. 1996. *Full Text Processing and Retrieval: Weight Ranking, Text Structuring, and Passage Retrieval for Arabic Documents*. Ph.D. thesis, Illinois Institute of Technology.

Hearst, Marti, Jan Pedersen, Peter Pirolli, Hinrich Schüetze, Gregory Grefenstette, and David Hull. 1996. Four TREC-4 Tracks: the Xerox site report. In Donna Harman, editor, *Proceedings of the Fourth Text Retrieval Conference TREC-4*. National Institute of Standards and Technology Special Publication. (to appear).

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

Hearst, Marti A. 1994a. *Context and Structure in Automated Full-Text Information Access*. Ph.D. thesis, University of California at Berkeley. (Computer Science Division Technical Report UCB/CSD-94/836).

Hearst, Marti A. 1994b. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Meeting of the Association for Computational Linguistics*, pages 9–16, Las Cruces, NM, June.

Hearst, Marti A. 1995. Tilebars: Visualization of term distribution information in full text information access. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, Denver, CO, May.

Hearst, Marti A. 1996. Improving full-text precision using simple query constraints. In *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval (SDAIR)*, Las Vegas, NV.

Hearst, Marti A. and Christian Plaunt. 1993. Subtopic structuring for full-length document access. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, pages 59–68, Pittsburgh, PA.

Hersh, William R., Diane L. Elliot, David H. Hickam, Stephanie L. Wolf, and Anna Molnar. 1995. Towards new measures of information retrieval evaluation. In *Proceedings of the 18th Annual International ACM/SIGIR Conference*, pages 164–170, Seattle, WA.

Hinds, John. 1979. Organizational patterns in discourse. In Talmy Givón, editor, *Syntax and Semantics: Discourse and Syntax*, volume 12. Academic Press, pages 135–158.

Hirschberg, Julia and Diane Litman. 1993. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics*, 19(3):501–530, Sept.

Hirschberg, Julia and Christine H. Nakatani. 1996. A prosodic analysis of discourse segments in direction-giving monologues. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 286–293, Santa Cruz, CA.

Hull, David and Gregory Grefenstette. 1995. Stemming algorithms – a case study for detailed evaluation. *Journal of the American Society for Information Science (JASIS)*, 46(9).

Hwang, Chung Hee and Lenhart K. Schubert. 1992. Tense trees as the 'fine structure' of discourse. In *Proceedings of the 30th Meeting of the Association for Computational Linguistics*, pages 232–240.

Isard, Amy and Jean Carletta. 1995. Replicability of transaction and action coding in the map task corpus. In Johanna Moore and Marilyn Walker, editors, *Empirical Methods in Discourse: Interpretation & Generation, AAAI Technical Report SS-95-06*. AAAI Press, Menlo Park, CA.

Karlgren, Jussi. 1996. Stylistic variation in an information retrieval experiment. In *Proceedings of the NeMLaP-2 Conference*, Ankara, Turkey, September.

Karttunen, Lauri, Kimmo Koskenniemi, and Ronald M. Kaplan. 1987. A compiler for two-level phonological rules. In Mary Dalrymple, editor, *Tools for Morphological Analysis*. Center for the Study of Language and Information, Stanford, CA.

Kozima, Hideki. 1993. Text segmentation based on similarity between words. In *Proceedings of the Student Session of the 31th Annual Meeting of the Association for Computational Linguistics*, pages 286–288, Columbus, OH.

Kuno, Susumo. 1972. Functional sentence perspective: A case study from Japanese and English. *Linguistic Inquiry*, 3(3):269–320.

Kupiec, Julian, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM/SIGIR Conference*, pages 68–73, Seattle, WA.

Lakoff, George P. 1972. Structural complexity in fairy tales. *The Study of Man*, 1:128–150.

Lewis, David D. and Philip J. Hayes. 1994. Special issue on text categorization. *Transactions of Office Information Systems*, 12(3).

Litman, Diane J. and Rebecca J. Passonneau. 1995. Combining multiple knowledge sources for discourse segmentation. In *Proceedings of the 33rd Meeting of the Association for Computational Linguistics*, pages 108–115, June.

Longacre, R. E. 1979. The paragraph as a grammatical unit. In Talmy Givón, editor, *Syntax and Semantics: Discourse and Syntax*, volume 12. Academic Press, pages 115–134.

Mann, William C. and Sandra A. Thompson. 1987. Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS 87-190, ISI.

Marchionini, Gary, Peter Liebscher, and Xia Lin. 1991. Authoring hyperdocuments: Designing for interaction. In Martin Dillon, editor, *Interfaces for Information Retrieval and Online Systems*. Greenwood Press, New York, NY, pages 119–131.

Miller, George A., Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to WordNet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244.

Mittendorf, Elke and Peter Schäuble. 1994. Document and passage retrieval based on Hidden Markov Models. In *Proceedings of the 17th Annual International ACM/SIGIR Conference*, pages 318–327, Dublin, Ireland.

Moffat, Alistair, Ron Sacks-Davis, Ross Wilkinson, and Justin Zobel. 1994. Retrieval of partial documents. In Donna Harman, editor, *Proceedings of the Second Text Retrieval Conference TREC-2*. National Institute of Standards and Technology Special Publication 500-215, pages 181–190.

Mooney, David J., M. Sandra Carberry, and Kathleen F. McCoy. 1990. The generation of high-level structure for extended explanations. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, volume 2, pages 276–281, Helsinki.

Moore, Johanna D. and Martha E. Pollack. 1992. A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics*, 18(4).

Morris, Jane. 1988. Lexical cohesion, the thesaurus, and the structure of text. Technical Report CSRI-219, Computer Systems Research Institute, University of Toronto.

Morris, Jane and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Nakatani, Christine H., Barbara J. Grosz, David D. Ahn, and Julia Hirschberg. 1995. Instructions for annotating discourses. Technical Report TR-25-95, Harvard University Center for Research in Computing Technology, Cambridge, MA.

Nomoto, Tadashi and Yoshihiko Nitta. 1994. A grammitoc-statistical approach to discourse partitioning. In *Proceedings of the Fifteenth International Conference on Computational Linguistics (COLING)*, pages 1145–1150, Kyoto, Japan, August.

Paice, Chris D. 1990. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management*, 26(1):171–186.

Passonneau, Rebecca J. and Diane J. Litman. 1993. Intention-based segmentation: Human reliability and correlation with linguistic cues. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 148–155.

Phillips, Martin. 1985. *Aspects of text structure: An investigation of the lexical organisation of text*. North-Holland, Amsterdam.

Resnik, Philip. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95)*, volume 1, pages 448–453, Montreal, Canada.

Reynar, Jeffrey C. 1994. An automateicmethod of finding topic boundaries. In *Proceedings of the Student Session of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 331–333, Las Cruces, NM.

Rosé, Carolyn Penstein. 1995. Conversation acts, interactional structure, and conversational outcomes. In Johanna Moore and Marilyn Walker, editors, *Empirical Methods in Discourse: Interpretation & Generation, AAAI Technical Report SS-95-06*. AAAI Press, Menlo Park, CA.

Rumelhart, David. 1975. Notes on a schema for stories. In Daniel G. Bobrow and Allan Collins, editors, *Representation and Understanding*. Academic Press, New York, pages 211–236.

Salton, Gerard. 1989. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley, Reading, MA.

Salton, Gerard and James Allan. 1993. Selective text utilization and text traversal. In *Proceedings of ACM Hypertext '93*.

Salton, Gerard, James Allan, and Chris Buckley. 1993. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, pages 49–58, Pittsburgh, PA.

Salton, Gerard, Amit Singhal, Chris Buckley, and Mandar Mitra. 1996. Automatic text decomposition using text segmentation and text themes. In *Proceedings of Hypertext '96, Seventh ACM Conference on Hypertext*, pages 53–65, Washington, D.C.

Schiffrin, Deborah. 1987. *Discourse Markers*. Cambridge University Press, Cambridge.

Schütze, Hinrich. 1993. Word space. In Stephen J. Hanson, Jack D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann, San Mateo CA.

Skorochod'ko, E.F. 1972. Adaptive method of automatic abstracting and indexing. In C.V. Freiman, editor, *Information Processing 71: Proceedings of the IFIP Congress 71*, pages 1179–1182. North-Holland Publishing Company.

Stark, Heather. 1988. What do paragraph markers do? *Discourse Processes*, 11(3):275–304.

Stoddard, Sally. 1991. *Text and Texture: Patterns of Cohesion*, volume XL of *Advances in Discourse Processes*. Ablex Publishing Corporation, Norwood, NJ.

Tannen, Deborah. 1989. *Talking Voices: Repetition, dialogue, and imagery in conversational discourse*. Studies in Interactional Sociolinguistics 6. Cambridge University Press.

Tombaugh, J., A. Lickorish, and P. Wright. 1987. Multi-window displays for readers of lengthy texts. *International Journal of Man [sic] -Machine Studies*, 26:597–615.

van der Eijk, Pim. 1994. Comparative discourse analysis of parallel texts. Technical Report cmp-lg/9407022, Digital Equipment Corporation.

van Dijk, Teun A. 1980. *Macrostructures*. Lawrence Erlbaum Associates, Hillsdale, N.J.

van Dijk, Teun A. 1981. *Studies in the Pragmatics of Discourse*. Mouton Publishers, The Hague.

Walker, Marilyn and Steve Whittaker. 1990. Mixed initiative dialogue: An investigation into discourse segmentation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 70–78.

Walker, Marilyn A. 1992. Redundancy in collaborative dialogue. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING)*, pages 345–351, Nantes, France, July.

Webber, Bonnie Lynn. 1987. The interpretation of tense in discourse. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 147–154, Stanford, CA.

Webber, Bonnie Lynn. 1988. Discourse deixis: Reference to discourse segments. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, pages 113–122, Buffalo, NY.

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*, pages 454–460, Nantes, France, July.

Youmans, Gilbert. 1991. A new tool for discourse analysis: The vocabulary-management profile. *Language*, 67(4):763–789.