DOCUMENT RESUME

ED 357 751 IR 054 559

AUTHOR Losee, Robert M., Jr.
TITLE A Model of Document Structure and an Application: Generating Hypertext from Linear Text.
PUB DATE 2 Mar 93
NOTE 19p.
PUB TYPE Reports - Evaluative/Feasibility (142)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Algorithms; Communication (Thought Transfer); Computer Assisted Instruction; Discourse Analysis; Feedback; *Hypermedia; Imagery; *Mathematical Models; Multimedia Instruction; Music; Tables (Data); User Needs (Information); *Writing (Composition)
IDENTIFIERS Document Analysis; *Document Structure; *Multimedia Materials; Passage Organization; Textual Analysis

ABSTRACT

A model of mono- or multimedia document structure is proposed consistent with the notion that there should be few changes in subject between one document passage and adjacent passages, where a passage may be a phrase, sentence, or paragraph. An algorithm for ordering passages in a composition is provided which is optimal given only a concern for the information theoretic similarity of adjacent passages. This procedure is easily coded and works well, given a set of independent subject-related features. This approach is applied to the transformation of linear text into hypertext. An expansion of the model has an economic component which allows for user feedback to modify the structure of the linear or hypertext documents. This model is media independent and may prove useful for analyzing and processing music, images, conversation, and social communication, as well as traditional text. Two figures and four tables illustrate the discussion. (Contains 22 references.) (Author/SLD)

A Model of Document Structure and an Application: Generating Hypertext from Linear Text

Robert M. Losee, Jr.
School of Information and Library Science
University of North Carolina
Chapel Hill, NC 27599-3360 U.S.A.

Phone: 919-962-7150
Fax: 919-962-8071
losee@ils.unc.edu

March 2, 1993

Abstract

A model of mono- or multimedia document structure is proposed consistent with the notion that there should be few changes in subject between one document passage and adjacent passages, where a passage may be a phrase, sentence, or paragraph. An algorithm for ordering passages in a composition is provided which is optimal given only a concern for the information theoretic similarity of adjacent passages. This procedure is easily coded and works well, given a set of independent subject-related features. This approach is applied to the transformation of linear text into hypertext. An expansion of the model has an economic component which allows for user feedback to modify the structure of the linear or hypertext documents. This model is media independent and may prove useful for analyzing and processing music, images, conversation, and social communication, as well as traditional text.

1 Introduction

As computers are increasingly used for the manipulation and analysis of text and multimedia documents, the development of formally justified methods for structuring documents, individually and collectively, is becoming more important. A model of document structure can be used as the basis for developing more sophisticated procedures that examine and manipulate documents as part of a system capable of improving human performance through the use of computers.

A simple model of document structure is proposed below, as well as a method provided for arranging a document in an "optimal" structure. Briefly, a set of text passages or fragment representations is ordered so that neighboring passages are highly "similar." This is based on the notion that there should not be large subject "jumps" between adjacent passages; text with such jumps is often seen as "choppy" by the reader. Similarity of subject is based here on the value of passage *features*, represented by binary values indicating the presence or absence of a word or other characteristic in a passage. For example, a document with only the first two features in the feature universe $\{dog, cat, gerbil\}$ would be represented by $\{1, 1, 0\}$. A document with a similar set of features would then be placed adjacent to this document under an optimal structuring. This structuring method works well in some circumstances (with human selected and assigned features) and fails to perform as well in others. This model may be applied to linear text, allowing it to be automatically converted to hypertext.

Document structures can exist on several different levels, allowing for different uses of the document. For example, documents can be examined

on the sentence, passage, theme, or physical unit levels. The nature of the display unit, the "smallest unit of text retrieved and displayed by the access software" (Girill, 1985, p. 355), obviously varies depending on the medium and presentation format desired. For example, childrens' "board books" seldom have more than a sentence per page, if they contain that much text. Hypertext systems usually have one or more paragraphs displayed in a window, the display unit, while libraries deal in whole volumes as the display units described in online catalogs.

Note that the term *document* is used here in a general sense and is intended to mean any form of expression, whether it is textual, musical, aural, or visual. Rather than including long lists of possible applications during the discussions below, the reader should assume that the author believes that this work is essentially media-independent, except when obviously not.

A definition of *document structure* might be as follows:

> The structure of a document is an ordering of a set of passages, each passage being represented by a list of its contents, providing, in effect, a statement of "what it is about."

One may now move further to define an *optimal* document structure:

> An *optimal* document structure is the ordering of a set of passages such that the "subject" distance between them is minimized.

Note that the answers to the questions "What should constitute subject distance?" or "How should the similarity of two document fragments be measured?" isn't obvious. In addition, there is a problem with minimizing the distance "between them;" between what documents should the distance be minimized? Should one minimize the average difference between adjacent documents, or should the average physical distances between *all* passages come into play, thus moving beyond adjacent documents?

In this article, we concentrate on the distances between adjacent documents. Further research may examine minimizing the degrees of subject-dissimilarity and distances between all passages. However, global optimization may yield counter-intuitive results on the small scale, something we wish to avoid when proposing an initial model whose function is as much to explain as it is to perform.

2 Ordering Passages & Documents

The subject of text and media are assumed her to be representable by sets of *features* or characteristic attributes. Each text string occurrence or bibliographic characteristic may be a feature. All features are subject bearing to

Decimal	*Binary*	*Gray Code*
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Table 1: Gray Code for numbers from 0 to 7.

some extent, indicating that the passage is "about" a certain topic to some degree, often by containing identifying information useful for showing relations, e.g., passages may be grouped by similar institution of origin. Such features are useful in matching passages or documents, such as in information retrieval, where documents are associated with a query, or as is done in hypertext systems, where passages are "linked" based on a relationship between passages that is perceived to exist by the users and authors. For purposes here, features will be assumed to be binary and take one of the two values, 1 or 0. Each features hold a position in a binary code, with the code or number's value being dependent on the value of each feature.

One particular code, referred to as the Gray code, is useful when attempting to structure documents or passages based on their characteristics. The binary Gray code provides a representation for each ordered item such that there is only 1 character difference between the representation for an item and the representation for the next item (Hamming, 1986). The *Hamming distance* between two individual binary representations for items is the number of features by which they differ (Losee, 1990). For example, the Hamming distance between 10101 and 11111 is 2 because the two representations differ in 2 positions, the second and the fourth. The Hamming distance is always 1 between the Gray code representations of two numbers, where one number is the increment of the other.

The *reflected Gray code* representations (Gilbert, 1958) used in this research for the numbers from 1 to 8 are given in Table 1. The pattern to this coding system may be seen by noting that a listing of a power of 2 number of codewords such as this can be split into two equal portions, with the top half having a 0 as the leftmost bit and the bottom half having a 1 as the leftmost bit, assuming the same number of bits in numbers in both halves

(Flores, 1956).

One can convert a standard binary number into the Gray code by moving from the rightmost bit to the leftmost bit of the standard binary number, changing the value of a bit if the bit to its left is a 1. A number's representation in Gray code can be easily changed into a standard binary number, again while moving from right to left, by changing the value of a bit if the sum of the bits to its left is an odd number.

An ordering procedure for representation and the represented passage can be implemented using the Gray code. Each profile is written as a number, with each position representing a feature's value. Using the reflecting Gray code in Table 1, a passage with profile 11 would be placed before one with characteristics 10, because the former precedes the latter in value.

The use of a Gray code based ordering system provides a linear ordering for all passages. While one could use a similarity measure to determine the relationships between representations (Losee, 1990), it would be necessary to compute similarity measures between each representation and each other representation when attempting to find the nearest neighbors. The use of the Gray code allows for the ordering to be done once for a given set of feature probabilities and costs or weights, with a sort of complexity approximately $O(\log_2 n)$.

Readers with only a weak interest in the probabilistic and economic underpinnings for this model may wish to skip the next two sections on a first reading.

3 Passage Distance and Feature Ordering

An examination of passage subject distance and dissimilarity may begin with treating the distance or difference between two identical features as 0 and the difference between two different features as 1. The expected dissimilarity between two passages for a given feature may be computed as the sum of the probabilities that the feature have different values given random ordering, i.e., $p_i(1 - p_i) + (1 - p_i)p_i$, where p_i is the probability feature i will have the value 1, indicating that the feature is present in the passage. Expected distance reaches its maximum when $p_i = .5$. We assume that the expected dissimilarity between adjacent passages is measured as the sum of the expected dissimilarity between features.

Ordering features before placing them into the Gray code may help minimize the expected distance between documents or passages (Losee, 1992). Features may be placed in any order into the Gray code, that is, different features may be arbitrarily assigned to different number positions. However, the average feature's expected dissimilarity is minimized if those features

with the least expected dissimilarity are placed furthest to the right in the code. The features with the greatest expected dissimilarity are placed to the left. Consider a set of fragments in a universe of 2 features: *history* and *hypertext*, the latter a less frequently occurring feature. By placing *history* to the left of *hypertext*, the average distance between adjacent passages is lower than would be obtained with the reverse ordering.

When counting in a binary number system, the rightmost bits "cycle" more frequently than bits to the left, resulting in a lower expected dissimilarity between adjacent passages than would be the case if the least probable features with the greatest expected dissimilarity were on the right side and cycled most frequently. When all features have low probabilities, they can be arranged from left to right in order of decreasing probability of occurrence, which is the same as ordering the features from left to right in decreasing order of their expected dissimilarity. However, in cases where some features have or might have probabilities over .5, ordering by decreasing expected dissimilarity, which is theoretically justified, will not produce the same ordering as ordering by decreasing probability. Most natural language terms occur in text with a probability of less than .1 and similar low probabilities might be expected of most other features, allowing these features to be ordered by decreasing probability.

4 An Economic Model for Passage Ordering

There is a cost or economic loss associated with having passages on the same subject at some distance from each other or, conversely, from having passages on dissimilar subjects adjacent to each other. Informally, a cost, denoted by C, is associated with features having different values in passages that are located less than an arbitrarily chosen distance apart. For purposes here, these costs will be assigned to features in passages (or the documents themselves) that are adjacent and have different values for the features. Thus, for a given feature, the cost or economic loss of having a 0 for that feature value in the first passage and a 1 for that feature in an adjacent, second fragment, is denoted as $C_{0,1}$, with a similar cost, $C_{1,0}$, for the first passage having a value of 1 and the second passage having a value of 0.

The *expected cost* of dissimilarity of a given feature, i, having a different value in one randomly selected passage than in a second randomly selected adjacent passage, is

$$p_i(1-p_i)C_{1,0} + (1-p_i)p_iC_{0,1} = p_i(1-p_i)(C_{1,0} + C_{0,1}),$$

the product of the expected dissimilarity and the cost associated with that

expected dissimilarity, where p_i is the probability that feature i has the value 1 in the database of passages. It is here assumed that $C_{0,0}$ and $C_{1,1}$ equal 0. For notational simplicity, the sum of costs $C_{1,0} + C_{0,1}$ is referred to as C_i for feature i. The expected cost of the difference between two immediately adjacent passages due to feature i is equal to

$$\delta_i = p_i(1 - p_i)C_i.$$

We will occasionally refer to "δ" values when it is not necessary to refer to a specific feature.

For the purpose of linking related documents or passages, indexing representations of passages are stored as binary vectors. Indexing features are numbered 1 through n, with feature n being at the rightmost ("least significant digit") side of the representation. It is assumed that space is left for the addition of new features, e.g. $n + 1$, $n + 2$, etc.

The cost of placing passages r and s immediately adjacent to each other, each with n binary features, denoted as $d_{r,1}, d_{r,2}, \ldots, d_{r,n}$ for passage r, with similar notation for passage s, is

$$\Delta_{r,s} = \sum_{i=1}^{n} |d_{r,i} - d_{s,i}| C_i,$$

assuming that features may be treated independently. This is the sum of costs for features which differ in value between passages r and s.

Text fragments are ordered so that the expected costs associated with an ordering of passages is made relatively small. This may be done through the ordering of features. Those features with the lowest δ values are placed furthest to the right in the number representing the passage's characteristics. If other feature values in the passage are identical, the passages with only the rightmost bit in the representation being different will be adjacent.

Features with higher expected cost of dissimilarities or δ values, on the other hand, will be placed on the left side of the "number" representing the fragment. All fragments having this feature will be grouped together and those fragments without this feature will be grouped together. The cost or loss associated with separating such features is minimized by placing them on the left, thus grouping the passages with the feature together.

5 Experimental Results

Several experimental tests of this model of document structure have been performed and point out some of the strengths and weaknesses of the approach. One set of tests compared the ability of this approach to arrange or

structure several collections of library books (Losee, 1992). These arrangements were then compared with the structure imposed by the existing library classification system. These similarity measurements were based on the number of similar subject headings that were assigned to the documents by the librarians. The application of out document structuring model to this data was found to place similar documents closer together than did the existing library classification system.

Another set of experiments has been conducted which applied this structuring model to individual sentences in a set of over 1000 article abstracts, which were extracted from a set of articles indexed by the phrase "Cystic Fibrosis" in the MEDLINE database (Wood et al., 1989). Features used in ordering sentences were the words occurring in the sentences. Sentences were ordered and then the expected number of abstracts with that ordering pattern were compared with the number of abstracts that actually had that pattern. A χ^2 test indicated that the orderings were significantly different; this was also rather obvious from an examination of the raw data. This may be interpreted as indicating that the orderings imposed by the model did not approximate the actual orderings provided by the human authors. Thus, the application of the document structuring model to this data using the feature sets described does not capture much of the underlying process used by humans in writing. Given the results from the first set of experiments described above, the author considers this failure to be due in whole or in part to the problems associated with using natural language terms in the passages as the primary subject bearing features describing the passages.

Parenthetically, the first and last sentences of abstracts were ordered so that they were adjacent in a "significant" number of cases, reflecting a strong subject similarity between these two sentences.

These abstract based experiments were run with a stoplist of 50 words available to exclude some common, non-subject related words. In one test, words on the stop list were used to exclude features from the set of features, while in another test, no words were excluded if the occurred in no more than half the documents. Both tests failed to provide the hoped for ordering.

These results suggest that the basic model does effectively structure and order when an appropriate set of features is available. It is also apparent that merely selecting all or most of the natural language terms in a sentence as its features results in poor structuring performance. Performance may be improved significantly if, through use of a procedure such as factor analysis (Deerwester et al., 1990; Borko, 1985), independent features or factors may be located and used in this ordering. These subject features would perfectly match the requirements of the structuring model and should perform well; they are approximated by the subject headings assigned by librarians to

books in the first experiment above.

6 Linear and Hypertext Documents

Hypertext systems have provided means by which text or media fragments (nodes), entire documents, or bibliographic records (Nelson, 1991) may be organized or structured through the addition of links, making the organization of the passages inherently non-linear. Note that the phrases "text fragments," "passages," and "documents" are used interchangeably here to represent anything which might be linked by a hypertext system or retrieved in an information retrieval system. A hypertext system may be understood as a series of text fragments that may be linked in any number of ways. Text fragments are displayed on a CRT in a display window, with each fragment in its own window. At any one time, there is exactly one window that is *active*, and material linked to the active window is displayed on the CRT. For purposes here, linked materials are assumed to be displayed on the screen with the active passage.

The linear structuring system proposed above has the capability to structure existing passages for display of related fragments, approximating many of the characteristics of hypertext systems. Describing hypertext links as representing user economic preferences furthermore allows for feedback to be incorporated into a hypertext system, providing a learning capability often missing in existing hypertext systems. While other weighting schemes have been proposed to support the processing of queries (Frisse, 1988; Losee, 1990), the weighting method used here is derived explicitly from information theoretic and economic considerations.

A document structured in a manner consistent with the proposed model will allow, after one passage has been placed on the screen, similar passages to be quickly placed in other windows on the screen without the system waiting for the user to request them or the user attempting to "link" to them.

Furnas (1986) suggests that a display might profitably contain "local detail and global context;" it should be noted that the linkless system described here displays what the user is interested in, be it local detail or global context. The display of text varying in detail and context can be obtained through the addition of *virtual features* (Losee, 1989), indicating features of a document such as ability to provide context or level of detail.

Text fragments may be stored or indexed in a linear array of representations ordered so that similar documents are placed adjacent to or near each other. This arrangement of passages to display from the linear array of passages may be based on adjacency, as above, or on the amount of weight a document has in a subject-bearing factor (Lelu, 1991). Wraparound between

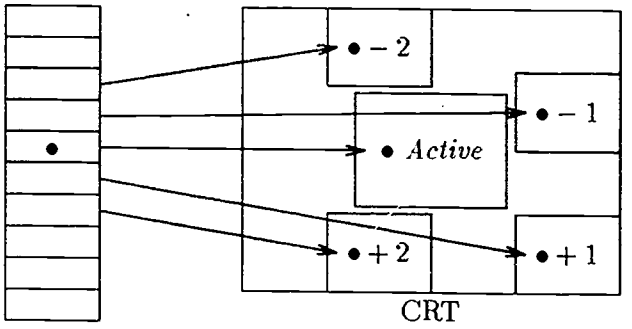

Figure 1: Hypertext display of a set of passages stored in a linear array.

the last item in the array and the first item in the array is assumed, so the last item is considered adjacent to the first item. Adjacent or neighboring passages are available for display when a link is made to a particular fragment by an action such as a mouse "click" on both fragments.

A sample screen is shown in Figure 1, illustrating how certain documents in a linear array of passages is displayed in a set of screen windows. Each window on the screen displays one of the elements of the one dimensional array (to the left) that is near the active window's location in the array.

An advantage of the method discussed here is that the order of text fragments is learnable, that is, the documents or text/media fragments can be arranged based upon relationships learned to be of interest and of economic benefit to the user. The addition of virtual features not directly present in a passage (Losee, 1989) to the set of independent features contained in passages can provide an additional basis for the relationship ordering decisions. The fewer are the number of links used in a hypertext system, then the closer will be the approximation of a hypertext system made by this linkless linear ordering system.

A link in a hypertext system may be represented by modifying the cost or economic loss for separating features. The cost estimate C_i may then be based on relationships between passages that the user has indicated are of interest.

Links in hypertext systems serve several purposes (DeRose, 1989). *Relational* links connect single locations for a variety of purposes. These include providing annotations, information about the text itself, as well as connections between existing passages.

Two approaches may be useful in representing hypertext links within the linkless hypertext system. One is to add a new feature, which is only contained in the directly linked fragments. This added feature may be place on the left or high order end of the fragment's representation. The documents with the value of 1 for this feature will be grouped together when ordered by the Gray code value, resulting in their eventual display together. The second means of representing a link is to modify the economic cost or weight for all features with the same values for both passages. By reordering the features into Gray code order, all text fragments containing these features will become ordered closer together.

A second family of hypertext links, *inclusion* links, connect a single passage or screen location to several target fragments. *Sequential* links require *ordered* target locations, including presentation as traditional linear text, while *taxonomic* links do not place the ordering requirement upon links.

The representation of inclusion links may take place by using a special group feature $\mathcal{G}_i$, containing a group of "regular" binary features of sufficient number to represent the data stored in the group-feature $\mathcal{G}$. The "subfeatures" within $\mathcal{G}$ may be ordered by placing them in a counting order. This provides the ordering for $\mathcal{G}$ values. By placing $\mathcal{G}$ to the left of the other features, the required ordering for emulating sequential links may be obtained. Taxonomic links may also be represented by a single feature common only to those linked fragments.

Intensional links are functionally based, providing links based upon the structure of a document and not primarily because of user implied individual interests. For example, links might exist between terms and their dictionary definitions. Representing intensional links requires the addition of a feature, e.g. *definition*, which is placed as feature $n+1$ on the right or low order end of the representation. This is because the cost of separating a fragment with the feature *definition* from another fragment, identical except for not having the feature *definition*, approaches 0. The value of the *definition* feature indicates whether the fragment constitutes a definition of another feature used in its representation. When fragments are ordered, those fragments which are identical except for the value of the added feature will be grouped together.

Once "links" have been created, it may be desirable to modify them. Feedback for our purposes consists of user supplied cost modifications. Indicating that there is a link between fragments F_r and F_s implies that the user anticipates an economic benefit from the existence of the link and from the passages being displayed together, with a loss associated with separating the text fragments.

Cost-based feedback can be saved as *link templates*, representing the in-

	Features				
	A	B	C	D	$\Delta_{2,-}$
F_1	0	0	0	1	2
F_2	0	1	0	0	
F_3	1	1	0	0	1
F_4	1	1	0	1	2
F_5	1	1	1	0	2
F_6	1	0	1	1	4
p_i	.5	.4	.3	.2	
C_i	1	1	1	1	
$p_i(1-p_i)C_i$	.25	.24	.21	.16	

Table 2: Sample passages with values for four features, A through D.

formation received from the user. This may be applied to the document or system at hand, as well as to future systems. By recording the relative costs of features, a template can be applied to other documents or systems, providing a customized ordering of passages and thus screen displays that benefit the end user who produced the feedback, as well as other, future users of this and other documents. Some link feedback may be best represented by a series of conditional rules within an expert system because of the complexity of the feature arrangements, e.g. those requiring the creation and placement of new features.

7 An Example

An example of the way in which feedback may be incorporated is as follows. The linkless hypertext system is initialized under the assumption that all links have equal economic worth, or equal costs, here arbitrarily treated as 1 unit of cost. This initialized system orders features and then orders passages based solely on the probabilities of features occurring. In this instance, this is the same as ordering the features and passages by the expected cost of the distance between adjacent passages.

A sample set of four features, A through D, as well as six passages or documents, F_1 through F_6, are provided in Table 2. The rarest feature, D, is on the right or least significant end while the most common feature, A, is on the left. Sorting the features in descending order by the $p_i(1-p_i)C_i$ values in the Table orders these features as they would be found if they were sorted by probability alone.

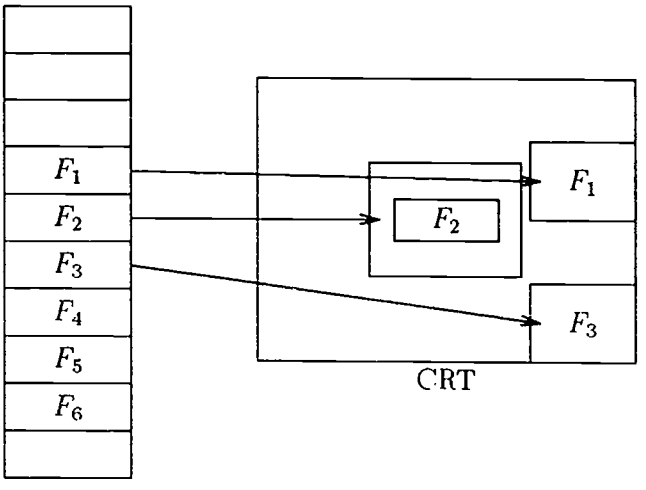

Figure 2: Display when F_2 is activated and no prior user feedback has been obtained.

Assume that three windows are available on a CRT screen. When passage F_2 is activated, the two neighboring fragments are displayed on the screen, as in Figure 2. The cost of passages being adjacent to passage F_2 are given at the right of Table 2. The two passages displayed with F_2 are those whose examination would result in the least cost to the user.

Assume now that the user has decided to "link" passages F_2 to F_5 after a search for fragments with a particular characteristic. It is thus desirable to make passages F_2 and F_5 adjacent. To do this, the cost associated with a feature with different values in passages F_2 and F_5 must be decreased. This results in a relative increase in the cost factor for features with identical values in the two adjacent documents, F_2 and F_5.

The system might use the following rule to perform feedback:

> *all features which are held in common by "linked" fragments should have their costs increased by a factor of* 10 *from their pre-feedback value.*

The cost of equal valued features increases because these features now have a greater "need" to be together for successful ordering. More formally, the

	Features				
	B	D	A	C	$\Delta_{2,-}$
F_6	0	1	1	1	22
F_1	0	1	0	0	20
F_4	1	1	1	0	11
F_3	1	0	1	0	1
F_5	1	0	1	1	2
F_2	1	0	0	0	
p_i	.4	.2	.5	.3	
C_i	10	10	1	1	
$p_i(1-p_i)C_i$	2.4	1.6	.25	.21	

Table 3: Features and passages after a link is established between F_2 and F_5.

increase in cost represents an increased loss associated with the presence of dissimilar feature values in adjacent documents.

Text fragments F_2 and F_5 have features B and D in common. Thus, the costs for B and D change from 1 to 10. When the features are rearranged in order of decreasing δ, they appear as in Table 3.

After linking F_2 and F_5, we might next arbitrarily link passages F_2 with F_3. Because these two fragments have values for features B, C, and D in common, the prior costs are multiplied by 10, resulting in the feature values given in Table 4.

8 Comparison with "Traditional" Hypertext

This set of examples illustrates both some strengths and weakness of applying the proposed document structure model to simulating hypertext using this linkless method. For example, based on theoretical considerations, passages or documents have been arranged and then presented on a screen so that linked or related documents are displayed together.

One drawback of this linkless method is that it cannot form links in all instances. Consider the linkage of passages F_2 and F_6, which may be seen as "opposites," that is, with no feature values in common. An attempt to link these two fragments by modifying costs will fail. The feature and passage orderings will remain the same as they were before the feedback was performed.

A second failure is that the model does not guarantee that recently linked passages always will be placed adjacent to the active passage. In Table 4,

	Features				
	B	D	C	A	$\Delta_{2,-}$
F_6	0	1	1	1	212
F_1	0	1	0	0	200
F_4	1	1	0	1	101
F_5	1	0	1	1	11
F_3	1	0	0	1	1
F_2	1	0	0	0	
p_i	.4	.2	.3	.5	
C_i	100	100	10	1	
$p_i(1-p_i)C_i$	24	16	2.1	.25	

Table 4: Features and passages after a link is established between F_2 and F_3.

for example, F_2 ideally should have appeared between F_3 and F_5 rather than between F_3 and F_6. An ad hoc procedure might be to display on the screen with the active passage those passages with the lowest Δ's.

This linkless method has several capabilities that would be difficult to implement in hypertext, although not impossible. Most important is the automatic incorporation of economic feedback into the model. This allows for the introduction of feedback that can be applied to any document the user examines. One's preferences can be easily stored and then applied to any document that this user might wish to examine. The linkless system also has the ability to essentially link *all* fragments with each other, and thus provide default links even though the user has not provided explicit feedback about relationships or established explicit links.

Hypertext allows numerous links between an active passage and other fragments. This can also be emulated by our system. although the linkless system would find it awkward to link fragments A and B and not link fragments A' and B', when fragments A and A' have almost identical features and B and B' have almost identical features. On the other hand, that hypertext allows this could, in fact, be considered a weakness of hypertext.

Another flaw of this linkless approach is that it assumes statistical independence of features. Dependence may be easily incorporated into this model by including features which represent joint feature occurrences. By using as many of these joint-feature occurrences as has a significant effect on system performance, system accuracy can be maximized while execution speed is minimized. Also. as presented above. the system is only concerned about binary feature occurrences. The linkless approach also assumes a stan-

dard fragment unit, such as the sentence or paragraph, while many hypertext systems allow links between various types of text units.

While it would be possible for this linkless approach to closely mimic any hypertext system, it might be awkward to do so. Empirical study of the large-scale use of hypertext in production environments will allow a determination to be made about the exact quantity and quality of links used for various categories of documents. This would allow a precise comparison of the performance of traditional hypertext and linkless systems.

9 Summary

All documents are structured in one form or another. A model of one type of document structure has been proposed and used successfully to provide such a structure. This model has then been applied to the structuring of non-linear documents. These non-linear documents can be seen to represent one form of this document structure model, and the transformation from linear to a non-linear form takes place within the constraints of the model.

Hypertext systems provide links by which users may access various parts of documents or whole documents. The links themselves represent interests or relationships that the user sees in the linked passages. The linkless method described here models document structure by explicitly assigning economic costs to links or relationships.

These costs may, in turn, be used in systems which order documents or passages so that an active window is displayed along with the documents whose relationships are most beneficial, given that particular active window. Because these costs are associated with specific relationships *per se* and not with specific passages, similar relationships may be assigned similar costs, allowing the system to "learn." Costs may be saved in templates, which may be retrieved and used with other document systems in future sessions.

The strongest attribute of this linkless model is that it provides an economic basis for understanding the links between documents or passages. It provides a generalization of some of the fundamental ideas underlying hypertext and allows for the simple introduction of these new functions into hypertext systems. Further work on this linkless approach to hypertext will examine means for incorporating feature dependencies into this model as well as comparing formal descriptions of this model with formal specifications of hypertext.

References

Borko, H. (1985). Research in computer based classification systems. In

Theory of Subject Analysis: A Sourcebook, pages 287–305. Libraries Unlimited, Littleton, Colo.

Cover, J. F. and Walsh, B. C. (1988). Online text retrieval via browsing. *Information Processing and Management*, 24(1):31–37.

Craven, T. C. (1989). An interactive aid for coding of sentence dependency structures. *Canadial Journal of Information Science*, 14(3):32–41.

Deerwester, S., Dumais, S. T., Furnas. G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*. 41(6):391–407.

DeRose. S. J. (1989). Expanding the notion of links. *SIGCHI Bulletin — Hypertext '89 Proceedings*, pages 249–257.

Flores, I. (1956). Reflected number systems. *IRE Transactions on Electronic Computers*, EC-5(2):79–82.

Frisse, M. E. (1988). Searching for information in a hypertext medical handbook. *Communications of the ACM*. 31(7):880–886.

Furnas (1986). Generalized fisheye views. In *Proceedings of the ACM CHI*. pages 16–23. ACM.

Gilbert, E. N. (1958). Gray codes and paths on the n-cube. *Bell System Technical Journal*, 37:815–826.

Girill, T. R. (1985). Narration, hierarchy, and autonomy: The problem of online text structure. *Proceedings of the 48th ASIS Annual Meeting*, 22:354–357.

Gray, S. H. and Shasha, D. (1989). To link or not to link? Empirical guidance for the design of nonlinear text systems. *Behavior Research Methods, Instruments, & Computers*. 21(2):326–333.

Hamming, R. (1986). *Coding and Information Theory*. Prentice-Hall, Englewood Cliffs, N.J., second edition.

Langford, D. (1990). Broadbutton node linking—A generalised approach to hyperbase navigation. *Hypermedia*. 2(2):159–169.

Lelu, A. (1991). Automatic generation of "hyper-paths" in information retrieval systems: A stochastic and an incremental algorithm. In *ACM Annual Conference on Research and Development in Information Retrieval*, pages 326–335, New York. ACM Press.

Losee, R. M. (1989). Minimizing information overload: The ranking of electronic messages. *Journal of Information Science*, 15(3):179–189.

Losee, R. M. (1990). *The Science of Information: Measurement and Applications.* Academic Press, New York.

Losee, R. M. (1992). A Gray code based ordering for documents on shelves: Classification for browsing and retrieval. *Journal of the American Society for Information Science*, 43(4):312–322.

Maeda, T. (1981). An approach toward functional text structure analysis of scientific and technical documents. *Information Processing and Management*, 17(6):329–339.

Marchionini, G. (1987). An invitation to browse. *Canadian Journal of Information Science*, 12(3/4):69–79.

McKnight, C., Dillon, A., and Richardson, J. (1991). *Hypertext in Context.* Cambridge University Press.

Nelson, M. J. (1991). The design of hypertext interfaces for information retrieval. *Canadian Journal of Information Science*, 16(2):1–12.

Wood, J. B., Wood, R. E., and Shaw, W. M. (1989). The cystic fibrosis database. Technical Report 8902, University of North Carolina, School of Information and Library Science, Chapel Hill, N.C.