The representation and comparison of hypertext structures using graph theory

Jonathan Furner
David Ellis, University of Sheffield
Peter Willett, University of Sheffield

Available at: https://works.bepress.com/furner/9/
THE REPRESENTATION AND COMPARISON OF HYPERTEXT STRUCTURES USING GRAPHS

Jonathan Furner*, David Ellis and Peter Willett

University of Sheffield (UK)

* Robert Gordon University (UK)

1 INTRODUCTION

The subject of this book is the hypertext retrieval system, which assists the user in the retrieval of information from collections of documents stored in the form of hypertext. Chapter 2 includes an overview of the structural and functional components of such a system, introducing the terms 'node' and 'link' in reference, respectively, to the representation of a document and the representation of a relationship between one document and another. In other chapters, more detailed descriptions are provided of the more important functions involved in the retrieval process, one of which may be identified as the task of link creation. In this stage, a representation, in the form of a link, is generated for each of the binary relationships existing between the documents in a collection. This activity may be carried out by automatic means (see Chapters 3 and 5 of this book), but is often undertaken manually—just as, in the task of document representation, index terms may be assigned to documents on the basis either of statistical computation or of human decision-making.

Our concern in the present chapter is with the logical structure of hypertext databases, rather than with any functional aspect of the retrieval system. Specifically, we are interested in the methods that may be used (i) in the representation of such structure, and (ii) in the comparison of representations of different structures. In Section 3, we describe how principles developed in the fields of graph theory and similarity measurement may be applied to these tasks. Firstly, however, we should explain the particular reasons we have for concerning ourselves with these matters; to this end, we outline in Section 2 the aims of the experiments in which we originally applied the methods described. The results of these experiments are briefly considered in Section 4, and ap
appropriate conclusions drawn in Section 5; full details are provided elsewhere [7, 8, 9, 10].

2 INTER-LINKER CONSISTENCY AND RETRIEVAL EFFECTIVENESS

One simple observation that may be made with respect to any set of $p$ nodes where $p > 2$, is that there exist $2^p$ possible sets of links that may be created amongst the nodes in that set. To appreciate the significance of such potential for variation, consider a hypothetical situation in which two sets of links are to be created amongst a single set of nodes, each on a separate occasion. If the links in each set were somehow to be generated randomly, then the probability of the two sets of links turning out to be equivalent would be dependent on the size of $p$, decreasing with increases in $p$. In real-life situations where linksets are created on the basis of human judgement—i.e., according to people’s subjective perceptions of the existence or otherwise of relationships between documents—the assignment of individual links is not a random matter; yet if the two sets of links in our hypothetical case were each to be created by a different person, we might reasonably conjecture that the likelihood of the resulting link-sets being equivalent, while not falling to the level observable in the random case, would still be small.

Support for such a conjecture may frequently be found in studies of a certain kind, common to a variety of disciplines, that report variation among the sets of judgements produced by different people in the course of conducting decision-making tasks. In the field of information retrieval, the most well-known of such studies are those dealing with inter-indexer consistency, whose results suggest that only a low level of agreement may generally be observed between the separate sets of index terms assigned to a single document by different indexers [17]. These studies have historically been considered significant on account of their common assumption that there exists a positive relationship between recorded levels of inter-indexer consistency and the levels of effectiveness that may be achieved by the retrieval system studied [18]. Accordingly, in some quarters, considerable thought has been applied to the task of suggesting ways in which levels of inter-indexer consistency (and hence, it is argued, levels of retrieval effectiveness) might be improved [12].

Underlying our experimental design was the contention that the manual creation of a set of links among the nodes of a hypertext database is sufficiently analogous to the manual assignment of a set of index terms to the documents in a conventional document database for it to be of interest to investigate the following hypothesis: that there exists a relationship between ‘inter-linker consistency’—i.e., the extent to which agreement exists in the choice of links to be created in a hypertext database—and the effectiveness of hypertext retrieval systems, similar to the one that is posited to exist between inter-indexer consistency and effectiveness. As Leonard [18] puts it, “the greater the agreement among indexers regarding the terms that best describe a document’s content, the higher the probability that the index terms will also match terms used in a search for which the document is regarded as a relevant item.” We hypothesised that it is similarly the case that the greater the agreement among link-creators regarding the links that best represent relationships between documents, the higher the probability that those links will be the ones which allow searches to navigate through the hypertext database with optimal effectiveness. We further surmised that, if we were to observe levels of inter-linker consistency that were both low and predictive of levels of retrieval effectiveness, then we might be led to conclude that steps should be taken to raise levels of consistency so that levels of effectiveness might be raised correspondingly.

In order to examine our hypothesis, we carried out an experiment in three stages as follows. In the first stage, we invited a number of people to create a set of links among the same set of nodes, each node representing a single paragraph of one of a collection of five full-text documents. The result was a test collection of twenty-five complete and separate hypertext ‘databases’—five hypertext versions of each of the five different documents, each sharing a common set of nodes with four others, but each having a possibly different inter-nodal link-set. We then calculated values representing the degree of similarity between the members of each pair of link-sets, and used these measurements in determining the mean level of inter-linker consistency characteristic of each database. In the second stage, we invited a number of people to carry out searches on a set of pre-determined queries in each of the databases. The result of each search was a list of nodes judged by the searcher to be relevant to the query, which could be compared with a corresponding list of nodes judged (before the search took place) by the document’s author to be relevant to that query. By means of such comparison, we calculated values representing the level of effectiveness of each search, and used these measurements in determining the mean level of retrieval effectiveness characteristic of each database. Finally, we were able to compare the two sets of scores, representing consistency and effectiveness respectively, in order to determine the nature of the relationship existing between them. The results of this analysis are briefly reviewed in Section 4. For the present, however, we are more interested in the methodology used in the first of the three stages outlined above.
3 THE MEASUREMENT OF INTER-LINKER CONSISTENCY: METHODOLOGY

Given our intention to calculate values that would indicate the extent of inter-linker consistency amongst the databases in our test collection, we set out to calculate a variety of values for each database-pair that might be said to represent the degree of similarity between the members of that pair. Methods for measuring levels of similarity have been described in many fields in which classification is an important endeavour—notably in numerical taxonomy (i.e. the study of the classification of biological species) [23] and both chemical [26] and textual [22] information retrieval. Lack of communication between these fields has resulted in much duplication of effort, but most would agree that the measurement of the degree of similarity between any two objects involves at least four steps. First, the objects that we wish to compare should be identified. Secondly, these objects should be represented in a way that allows us to calculate numerical values quantifying the degree of similarity between them. Thirdly, a set should be selected of those attributes of these object-representations whose values are to be used to characterise the representations in the course of their comparison. At this stage, a weighting scheme may be implemented that emphasises certain attributes according to any perceived differences in their relative significance. Fourthly, we need to select an appropriate measure or coefficient, whose values are to be derived from an analysis of the various attribute values of the two object representations, and used to indicate the level of similarity between them. Accordingly, the measurement of the degree of similarity between the objects in our study involved four steps, each of which is described in turn below.

3.1 The Identification of Objects for Comparison

The objects that we wished to compare were the five databases created from each of the five original full-text documents. Each database could be compared with four others, resulting in ten database-pairs per document.

3.2 The Numerical Representation of Objects

Our aim at this stage, to represent each hypertext database in a way that would allow us to calculate numerical values quantifying the degree of similarity between them, may be identified as an instance of a more general aim—to form conceptual representations, or models, of the logical structure of hypertext databases, so that their characteristics may be studied more easily. By referring to ‘structure’, we focus on the objective and relational components of hypertext retrieval systems, rather than on their functions; similarly, by referring to ‘logical’ structure, we establish our concern with the definition of mathematical sets of objects and relations, rather than with the specification of arrangements of cells in which these elements may physically be stored in a computer. It is possible to model the logical structure of any real-world system using concepts developed in the branch of mathematics known as graph theory [14]. Techniques based on graph-theoretic concepts have thus been applied in very many different fields: here, we provide a brief introduction to the specific ways in which such techniques may be applied in the field of hypertext. (Much of the terminology used in the following account derives from set theory: for explanations of unfamiliar terms, the interested reader should consult a basic textbook in that subject.)

The Directed-Graph Model of Hypertext

A graph may be defined as an ordered pair of the form $G = (V, E)$, where $V$ is a datum set given by $V = \{v_1, v_2, v_3, \ldots, v_p\}$, and $E$ is a family of two-member subsets of $V$ given by $E = \{(v_i, v_j) \mid (v_i, v_j), \ldots, (v_i, v_j)\}$. Each of the $p$ members of $V$ is called a vertex, and each of the $r$ members of $E$ is called an edge. Each edge is said to ‘join’ the two vertices that are its members. If all its vertices are distinguishable from one another by unique names such as $v_1$ and $v_2$, a graph is said to be labelled.

The logical structure of a hypertext database may be modelled by the ordered pair $D = (V, E)$, where $V$ is a datum set given by $V = \{v_1, v_2, v_3, \ldots, v_p\}$, each of whose $p$ members is a labelled vertex representing a document, and $E$ is a relation set given by $E \subseteq V \times V$, each of whose $r$ members is an ordered pair (i.e., a ‘directed’ edge) representing a relationship between two documents. In other words, $V$ has the same denotation as it does in the definition given above for a graph, whereas $E$ denotes a set of ordered pairs of members of $V$ rather than a set of non-ordered such pairs. Each vertex in this model corresponds to a node in the database, and each directed edge corresponds to a link.
Ordered pairs such as \( D \) are known as directed graphs or digraphs: they are graphs in which each edge 'points' in some direction. Given \( e_k = (v_i, v_j) \), we can say that \( e_k \) 'starts' at its first co-ordinate or source \( v_i \), 'finishes' at its second co-ordinate or target \( v_j \), and 'points' from \( v_i \) to \( v_j \). If two vertices in a directed graph \( v_i \) and \( v_j \) are related by an edge pointing from \( v_i \) to \( v_j \), then \( v_j \) is said to be adjacent to \( v_i \), but \( v_i \) is not considered adjacent to \( v_j \) unless a corresponding edge also exists that points from \( v_j \) to \( v_i \). The vertex \( v_j \) may also be said to be a child of \( v_i \), and \( v_i \) a parent of \( v_j \). A path in a directed graph is any sequence of vertices and edges of the form \( v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow \ldots \), where \( v_j \) is adjacent to \( v_i \), \( v_k \) is adjacent to \( v_j \), and so on. The length of a path is equal to the number of edges in its sequence. Where there exists a path starting at \( v_i \) and finishing at \( v_j \), \( v_i \) is said to be an ancestor of \( v_j \), and \( v_j \) a descendant of \( v_i \). A path is called a cycle if its edges and vertices are all different except for its starting and finishing vertices which are the same. A graph is acyclic if it has no cycles: directed acyclic graphs (DAGs), in which no ancestor of a vertex is also a descendant of that vertex, are commonly used to model hypertext databases [20].

Awareness of the significance of network-like structures in hypertext may be traced back at least to the work of Nelson in the 1960s [19], but analysis of such structures in terms of formal graph-theoretic concepts is a more recent development. In their discussion of the Neptune system, Deisile and Schwartz [5] were among the first to make explicit use of the language of graph theory: “If the nodes and links of a hyperdocument are mapped in the obvious way to nodes and edges of an abstract graph, then a hyperdocument can map onto an arbitrary graph (with the possibility of cycles) called a hypergraph.” Since the mid-1980s, several other logical models have been developed that provide, in far greater depth, a formal analysis of the functions specifying the transformation of objects and relations that may take place during the retrieval process (see, for example, Halasz and Schwartz’s definition of the Dexter hypertext reference model [13], which has been used as a template in the design of a variety of systems); however, the basic directed-graph model remains fundamental to much of the published work on hypertext structure.

**The Representation of Graphs by Matrices**

Graphs may easily be represented using the language of set theory, as above, or diagrammatically, using pictures made up of points and lines. However, in order that a graph may be analysed by computer, it is necessary for it to be represented in a machine-readable format, which normally involves some form of numerical matrix. Examples are the adjacency, distance and converted distance matrices.

For any graph \( G \) of \( p \) vertices, we can define an adjacency matrix \( A(G) \) consisting of \( p \times p \) elements of the form \( a_{ij} \). Each of these elements has a value equal to 1 (if the vertex \( v_j \) is adjacent to the vertex \( v_i \)), 0 (if \( v_j \) is not adjacent to \( v_i \)) or null (if \( i = j \), i.e., if \( v_i \) and \( v_j \) denote the same vertex). The out-degree \( i \) of a vertex \( v_i \) is equal to the number of vertices that are adjacent to \( v_i \), and may be calculated from an adjacency matrix by summing all the values in the \( i \)th row. The in-degree of a vertex \( v_j \) is equal to the number of vertices to which \( v_j \) is adjacent, and may be calculated from an adjacency matrix by summing all the values in the \( j \)th column. In a non-directed graph, the out-degree of a vertex is equal to its in-degree: this value is usually known simply as its degree.

We can also define a distance matrix \( D(G) \) consisting of \( p \times p \) elements of the form \( d_{ij} \). Each of these elements has a value equal to the length (measured in links) of any shortest path starting at \( v_i \) and finishing at \( v_j \) (if those vertices are connected), 0 (if those vertices are not connected), or null (if \( i = j \)). The status (or distance-sum) \( s_i \) of a vertex \( v_i \) is equal to the sum of the lengths of the shortest paths from \( v_i \) to every other vertex, and may be calculated from a distance matrix by summing all the values in the \( i \)th row. The contraststatus of a vertex \( v_j \) is equal to the sum of the lengths of the shortest paths from every other vertex to \( v_j \), and may be calculated from a distance matrix by summing all the values in the \( j \)th column. In a non-directed graph, the status of a vertex is equal to its contraststatus: this value is usually known simply as its distance-sum.

Botafogo et al. [4] introduce the idea of a converted distance matrix, which we may represent as \( D'(G) \) and defined as consisting of \( p \times p \) elements of the form \( d'_{ij} \), where \( d'_{ij} \) is equal to \( d_{ij} \) if \( d_{ij} \neq 0 \) or \( p \) (if \( d_{ij} = 0 \)). Botafogo et al. argue that the distance between two unconnected vertices is nearer infinity than is indicated by the value 0, but, given the difficulty of handling infinite values, they suggest that each such value should be replaced with a value that is equal to \( \max(d_{ij}) + 1 \), where \( \max(d_{ij}) \) is equal to the maximum value \( d_{ij} \) might possibly take (i.e., \( p - 1 \)). Converted distance matrices are useful representations only of directed graphs or non-connected non-directed graphs: the converted distance matrix of a connected non-directed graph is identical to its distance matrix. The converted out-distance \( s_i \) of a vertex \( v_i \) may be calculated from a converted distance matrix by summing all the values in the \( i \)th row; the converted in-distance of a vertex \( v_j \) may be calculated by summing all the values in the \( j \)th column.
### 3.3 The Selection of Attributes

The sets of elements contained in the matrical representations of our hypertext databases were thus identified as the object-representations that we wished to compare. The next step involved the selection of attributes of these matrical representations whose values were to be used to characterise the representations in the course of their comparison. Any set of attribute-values may be expressed in the form of an n-tuple, where n is the number of attributes whose values are used to characterise the object-representation. We may derive n-tuples of attribute-values from any graph-theoretic representation of a hypertext database \( G \) with \( p \) nodes in two general ways as follows.

#### Matrix-Element Method

This method involves using the elements of the matrix directly as attribute-values. We can do this either (i) by representing the matrix as a single n-tuple (where \( n = p^2 - p \)), or (ii) by representing the matrix as an n-tuple (where \( n = p \)) of n-tuples (where \( n = p - 1 \)). In other words, the matrix \( X(G) \) (where each element of the matrix is given by \( x_{ik} \)) may be considered either (i) as a single n-tuple \( X_1 \), or (ii) as a set of \( p \) n-tuples \( X_i \). For instance, the adjacency matrix for the example hypertext database \( G_1 \) shown in Fig. 1 may be represented either by the single 20-tuple \( X_1 = (1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \), or by the five 4-tuples \( X_{1,1} = (1, 1, 0, 0), X_{2,1} = (0, 0, 1, 0), X_{3,1} = (0, 0, 1, 0), X_{4,1} = (0, 0, 0, 0) \) and \( X_{5,1} = (0, 0, 0, 0) \). Examples of the use of the latter method of characterising graphs may be found in the fields of chemical information retrieval (in the analysis of augmented-atom fragments [26]) and cognitive science (in the analysis of ‘pathfinder’ network models [11]).

#### Topological-Index Method

The manipulation of matrices is particularly computationally expensive, and much research has been undertaken with the aim of identifying properties of graphs that may be used to represent those graphs in lieu of full matrices. This work has historically been the domain of computational chemists, who use the vertices and edges of graphs to model the atoms and bonds of molecules, and various formulæ have been suggested in this field for the calculation of values of attributes known as topological indices [2]. A topological index is an attribute of a graph that satisfies the following requirements:
Its values should be derived from the graph’s topological (i.e., relational) characteristics.

It should have a low level of degeneracy—i.e., it should be able to discriminate between as many different graphs as possible. Ideally, the attribute would be a ‘complete graph invariant’, each of its values being unique to an individual structure.

Its values should be simple in form, such as single numbers or short sequences of numbers, and should be easy to compute.

The algorithms used in the computation of its values should be efficient. In practical terms, an algorithm may be considered efficient if it returns results within an acceptable period of time; this is usually the case if the number of operations required by the algorithm increases only polynomially (rather than exponentially or worse) in relation to any increase in the size of the graph.

The simplest of topological indices are those defined by functions on the data elements of adjacency, distance and converted matrices. For our present purposes, we can identify two classes of such indices: a class of ‘node’ indices, the values of whose members take the form of an n-tuple of numerical elements and a class of ‘graph’ indices, the values of whose members take the form of single numbers. There are very many topological indices whose values can be calculated for a graph and the formulae of a selection of members of each class are summarised in Tables 1 and 2. It will be appreciated that the simpler of these indices do not exhibit a particularly low level of degeneracy. $G3(D)$ is known as the compactness of a graph $G$, $G4(D)$ as the absolute prestige sum of $G$, $G5(D)$ (where $L = \frac{k^2}{2}$ if $p$ is even, or $L = \frac{k(k-2)}{4}$ if $p$ is odd) as the stratum of $G$, $G6(A)$ and $G6(D)$ (where the sum is over all adjacent vertices) as the connectivity index and distance-sum connectivity index of $G$ respectively, and $G7(A)$ and $G7(D)$ (where $r = \text{the number of edges and } p = \text{the number of vertices}$) as the ‘mean’ connectivity index and ‘mean’ distance-sum connectivity index of $G$ respectively.

**Node Indices**

The concepts of the out-degree ($N1(A)$), in-degree ($N2(A)$), status ($N1(D)$), contrastatus ($N2(D)$), converted in-distance ($N1(D')$) and converted out-distance ($N2(D')$) of a vertex were introduced in our discussion of matrices in Section 3.2. Out-degree, in-degree, status, contrastatus and prestige ($N3(D)$) are commonly employed in analyses of directed graphs (see, for example, Harary et al. [15]). Converted distances and relative out-centrality ($N4(D')$) are discussed by Botafoho et al. [4]; texture ($N5(D)$) is defined by Bernstein et al. [3].

<table>
<thead>
<tr>
<th>Matrix type</th>
<th>Adjacency</th>
<th>Distance</th>
<th>Converted Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(i,j)$</td>
<td>$a_{ij}$</td>
<td>$d_{ij}$</td>
<td>$d'_{ij}$</td>
</tr>
<tr>
<td>$N1$</td>
<td>$\sum_{i=1}^{n} a_{ij} = \delta_i$</td>
<td>$\sum_{j=1}^{n} d_{ij} = s_i$</td>
<td>$\sum_{j=1}^{n} d'_{ij} = s'_i$</td>
</tr>
<tr>
<td></td>
<td>Out-degree</td>
<td>Status (or distance-sum)</td>
<td>Converted out-distance</td>
</tr>
<tr>
<td></td>
<td>of $v_i$</td>
<td>of $v_i$</td>
<td>of $v_i$</td>
</tr>
<tr>
<td>$N2$</td>
<td>$\sum_{i=1}^{p} a_{ij}$</td>
<td>$\sum_{i=1}^{p} d_{ij}$</td>
<td>$\sum_{i=1}^{p} d'_{ij} = s'_i$</td>
</tr>
<tr>
<td></td>
<td>In-degree</td>
<td>Contrastatus</td>
<td>Converted in-distance</td>
</tr>
<tr>
<td></td>
<td>of $v_j$</td>
<td>of $v_j$</td>
<td>of $v_j$</td>
</tr>
<tr>
<td>$N3$</td>
<td>$\sum_{j=1}^{n} a_{ij} - \sum_{i=1}^{n} a_{ij}$</td>
<td>$\sum_{j=1}^{n} d_{ij} - \sum_{i=1}^{n} d_{ij}$</td>
<td>$\sum_{j=1}^{n} d'<em>{ij} - \sum</em>{i=1}^{n} d'_{ij}$</td>
</tr>
<tr>
<td></td>
<td>Prestige (net status)</td>
<td>Prestige (net status)</td>
<td>Prestige (net status)</td>
</tr>
<tr>
<td></td>
<td>of $v_i$</td>
<td>of $v_i$</td>
<td>of $v_i$</td>
</tr>
<tr>
<td>$N4$</td>
<td>$\sum_{i=1}^{p} s_{ij}$</td>
<td>$\sum_{i=1}^{p} s_{ij}$</td>
<td>$\sum_{i=1}^{p} s'_{ij}$</td>
</tr>
<tr>
<td></td>
<td>Relative out-centrality of $v_i$</td>
<td>Relative out-centrality of $v_i$</td>
<td>Relative out-centrality of $v_i$</td>
</tr>
<tr>
<td>$N5$</td>
<td>$\sum_{j=1}^{n} \frac{1}{d_{ij}}$</td>
<td>$\sum_{j=1}^{n} \frac{1}{d'_{ij}}$</td>
<td>$\sum_{j=1}^{n} \frac{1}{d'_{ij}}$</td>
</tr>
<tr>
<td></td>
<td>Texture</td>
<td>Texture</td>
<td>Texture</td>
</tr>
<tr>
<td></td>
<td>of $v_i$</td>
<td>of $v_i$</td>
<td>of $v_i$</td>
</tr>
</tbody>
</table>

**Table 1** Formulea for Node Indices
who established that the numbers of paths in graph-theoretical structures could provide a useful basis for the derivation of topological indices. The value of his index is equal to the sum of the distances (i.e., the lengths in edges of any shortest path) between each pair of vertices in a graph, and may be calculated from the distance matrix using the formula:

\[ W = \frac{1}{2} \sum_{i=1}^{p} s_i \]

where \( s_i \) is the distance-sum of the vertex \( v_i \), and \( p \) is the number of vertices in the graph [16]. \( W \) is thus equal to \( G2(D) \).

Because molecules are represented by graphs that are non-directed, in its typical applications the values of the Wiener index may be derived from analysis of the upper triangular submatrix of the appropriate matrix (i.e., from that portion of the matrix above and to the right of its leading diagonal). The value of the non-normalised Wiener index varies in positive relation to the number of vertices in a graph. Values may alternatively be calculated for a ‘mean’ Wiener index (\( G2(D) \)) using a formula that is applicable to directed graphs, and that is normalised by the number of vertices in the graph:

\[ G2(D) = \frac{\sum_{i=1}^{p} s_i}{p} \]

The Wiener index is simple in form, and its values easy to calculate, but it is far from being a complete graph invariant: there are many pairs of graphs that are non-isomorphic but share the same value of \( W \) because they share the same distribution of distances. The more complex connectivity indices, which are based on the assignment of specific weights to different edges, are less degenerate. For example, the values of Randić’s index [21] may be calculated from the adjacency matrix using the formula:

\[ \chi = \sum (\delta_i \delta_j)^{-\frac{1}{2}} \]

where \( \delta_i \) is the out-degree of the vertex \( v_i \). \( \chi \) is recorded in Table 2 as \( G6(A) \). The summation is over all adjacent vertices \( v_i \) and \( v_j \) (i.e., over all paths of one edge in length). The value of the non-normalised Randić index varies in positive relation to the number of edges in a graph. Values may alternatively be calculated for a ‘mean’ Randić index (\( G7(A) \)) using a formula that is normalised by the number of edges, or pairs of adjacent vertices, in the graph:

\[ G7(A) = \frac{\sum (\delta_i \delta_j)^{-\frac{1}{2}}}{r} \]

---

### Table 2 Formulae for Graph Indices

<table>
<thead>
<tr>
<th>Matrix type</th>
<th>Code</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Adjacency</strong></td>
<td>( G1(A) )</td>
<td>( \sum_{i=1}^{p} s_i )</td>
</tr>
<tr>
<td><strong>Distance</strong></td>
<td>( G1(D) )</td>
<td>( \sum_{i=1}^{p} s_i )</td>
</tr>
<tr>
<td><strong>Conv. Distance</strong></td>
<td>( G1(D') )</td>
<td>( \sum_{i=1}^{p} s'_i )</td>
</tr>
<tr>
<td><strong>Adjacency</strong></td>
<td>( G2(A) )</td>
<td>( \sum_{i=1}^{p} s_i = \mu )</td>
</tr>
<tr>
<td><strong>Distance</strong></td>
<td>( G2(D) )</td>
<td>( \sum_{i=1}^{p} s_i )</td>
</tr>
<tr>
<td><strong>Conv. Distance</strong></td>
<td>( G2(D') )</td>
<td>( \sum_{i=1}^{p} s'_i )</td>
</tr>
<tr>
<td><strong>Conv. Distance</strong></td>
<td>( G3(D') )</td>
<td>( \frac{p^3 - p^2 - \sum_{i=1}^{p} s'_i}{p^3 - 2p^2 - p} )</td>
</tr>
<tr>
<td><strong>Distance</strong></td>
<td>( G4(D) )</td>
<td>( \sum_{h=1}^{p} \left</td>
</tr>
<tr>
<td><strong>Distance</strong></td>
<td>( G5(D) )</td>
<td>( \sum_{h=1}^{p} \left</td>
</tr>
<tr>
<td><strong>Adjacency</strong></td>
<td>( G6(A) )</td>
<td>( \sum (\delta_i \delta_j)^{-\frac{1}{2}} = \chi )</td>
</tr>
<tr>
<td><strong>Distance</strong></td>
<td>( G6(D) )</td>
<td>( \sum (s_i s_j)^{-\frac{1}{2}} )</td>
</tr>
<tr>
<td><strong>Adjacency</strong></td>
<td>( G7(A) )</td>
<td>( \frac{\chi}{r} )</td>
</tr>
<tr>
<td><strong>Distance</strong></td>
<td>( G7(D) )</td>
<td>( \frac{\sum_{i=1}^{p} (s_i s_j)^{-\frac{1}{2}}}{r} = J )</td>
</tr>
</tbody>
</table>

**Graph Indices**

A graph index used widely in computational chemistry is the Wiener index, \( W \). Although he did not use the language of graph theory, it was Wiener [20]...
where $r$ is the number of edges in the graph [24]. The ‘mean’ distance-sum connectivity index $J$ [1], recorded in Table 2 as $G7(D)$, is calculated from $G6(D)$ in a similar, but not identical, way to that in which $G7(A)$ is calculated from $G6(A)$. The value of the non-normalised index $G6(D)$ varies both in inverse relation to the number of edges in a graph, and in positive relation to the number of cycles in that graph. Both the number of edges and the number of cycles can be expressed simply by single values, and the normalising factor by which $G6(D)$ is multiplied in the formula for $G7(D)$ is designed to ensure that values of $G7(D)$ vary according to topological characteristics that are less easily isolated.

The compactness ($G3(D')$) and stratum ($G5(D)$) of a graph are discussed by Botafogo et al. [4]. The values of both of these indices, derived from converted distance and distance data respectively, vary between 0 and 1. A compactness value of 1 indicates a completely connected graph (i.e., one in which every vertex is linked to every other vertex), while 0 indicates a completely disconnected graph (i.e., one in which no vertex is linked to another). A stratum value of 1 indicates a perfectly linear topology (i.e., a chain whose edges may be traversed only in one order), while 0 indicates a completely cyclical topology (i.e., a ring whose edges may be traversed in as many orders as there are vertices).

Finally, another simple (and far from graph-invariant) index that may be used to characterise hypertext databases is one that does not even require their representation by matrices, and is given simply by $\frac{n}{c}$ (i.e., the ratio of the number of nodes to the number of links). Parunak [20] is among those who have recommended the use of this index for the characterisation of hypertext topologies.

Figure 2 presents values, for the same two graphs as represented earlier in Fig. 1, of each of the node and graph indices discussed above. Once again, the upper set of matrices and index values relates to $G_1$, the lower set to $G_2$. In our experiments, in order to measure the degree of similarity between hypertext databases, we characterised each database both (i) by the $n$-tuples of attribute-values (where $n = p$, the number of nodes) made up by the values of each of the node indices defined in Table 1, rather than of the data elements of a matrix, or (ii) by the single values of each of the graph indices defined in Table 2.

### 3.4 Selection of Coefficients

In this final step, we needed to select an appropriate measure or coefficient, whose values were to be derived from analysis of the various attribute-values of the two object-representations in a pair. If the objects to be compared are characterised solely by a single attribute (e.g., a graph index) whose values took the form of single numbers, it is a simple task to calculate the numerical difference between two such values, and thus to derive a value for a primitive measure of similarity: the smaller the difference, the more similar are the objects. If, however, the objects are each characterised by an attribute whose values take the form of sequences of numbers, or by more than one attribute, then one of a number of more complex arithmetic functions must be used in order to calculate a value for a measure of the agreement between the sets of
attribute-values of a pair of objects. These functions are known as similarity coefficients: whatever the nature of the attribute-values used, we may use any of a wide variety of such coefficients to compare a pair of n-tuples. Elsewhere, we have reviewed the composition and role of a variety of similarity coefficients [6], and in the course of our experiments we calculated sets of similarity values using no less than twenty-seven different formulae.

It is instructive in the current context to expand on issues relating to the equivalency and monotonicity of the similarity coefficients that we used in our experiments. The mass of results produced by calculations involving twenty-seven different coefficients was indigestible in itself. But in order to focus more clearly on the most informative data, we were able to eliminate redundancies using test statistics in the following manner. We constructed a series of correlation matrices, each cell of which contained the value of Pearson’s product-moment correlation coefficient $r$ derived from comparison of two sets of data, viz. those values of two different similarity coefficients contained in a corresponding table of source data. It is commonly noted (i) that some coefficients, identified by different names, are represented by formulae that are arithmetically equivalent, and (ii) that some coefficients are jointly monotonic with another, i.e., that it can be shown that the ranking of all measurements of similarity between pairs of objects in a specific sets is the same using one coefficient as it is using the other [6]. Inspection of the values in these correlation matrices allowed us:

- to confirm the arithmetic equivalence (for binary data) of:
  - the mean Manhattan metric, the mean squared Euclidean distance and the mean Canberra metric,
  - the complement of the mean Manhattan metric, the simple matching coefficient and the converted Hamann coefficient, and
  - the complement of the Bray/Curtis coefficient and the Dice coefficient;

- to confirm the joint monotonicity (for binary data, at a level of $r > 0.94$) of:
  - the Tanimoto, Dice, Sokal/Sneath (1), Cosine, Fossum and Pearson coefficients, and
  - the mean Manhattan, mean squared Euclidean and mean Canberra metrics, and the divergence, simple matching, Hamann, Sokal/Sneath (2) and Rogers/Tanimoto coefficients;

- to confirm the joint monotonicity (for non-binary data, at a level of $r > 0.94$) of:
  - the Tanimoto, Dice, Sokal/Sneath (1), cosine and Bray/Curtis coefficients,
  - the mean Manhattan, and mean squared Euclidean metrics, and
  - the mean Canberra metric and the divergence coefficient.

In practice, we were able to avoid unnecessary duplication of results by considering the values of only one coefficient from each of these clusters. Moreover, we were able to argue that the use of any representative from certain clusters would be inappropriate for other reasons: the weighting, mainly by distance coefficients, of negative matches and positive matches at the same level; the requirement, mainly by correlation coefficients, for the attribute-values under consideration to form a normal distribution; and unusually unwieldy ranges of values.

Most applications in information retrieval display a historical attachment to simple, linear, non-probabilistic association coefficients such as those provided by the Dice and Cosine formulae, and our findings suggested no reason to break with tradition. Association coefficients are based on a function known as the inner product, i.e., the sum of the products of corresponding elements in a pair of n-tuples, $X_a$ and $X_b$. The characteristic by which coefficients of this type may be distinguished is the composition of the factor by which they require the inner product to be multiplied. This normalisation ensures that the values of the coefficients remain within a specific range, such as that bounded by 0 and 1, with greater similarity being indicated by higher values. The Dice coefficient is given by:

$$S_D = \frac{2 \sum (x_{ij} y_{ij})}{\sum (x_{ij})^2 + \sum (y_{ij})^2}$$

and the Cosine coefficient by:

$$S_C = \frac{\sum (x_{ij} y_{ij})}{\sqrt{\sum (x_{ij})^2 \sum (y_{ij})^2}}$$

where the summation in all cases is over $j = 1$ to $j = n$. 
4 RESULTS

As we indicated in Section 2, the databases used in our experiments derived from five printed full-text documents, each a thesis, journal article or book written by a member of the Department of Information Studies at the University of Sheffield. Each printed document comprised a set of paragraphs, ranging in number from 23 to 347, arranged in a linear sequence. A machine-readable version of each of the documents was produced using the authoring system Guide. Each of these electronic documents consisted of a set of nodes (or ‘frames’ in Guide-speak), each node containing a single paragraph from the original printed document.

Copies of each of the five electronic documents were allocated to each of five student volunteers. Each of the five volunteers had been instructed in the use of an interactive system, developed using Guide, that allowed them to create explicit representations of links between paragraphs whose contents they decided were related. On completion of the linkers' work, the results were five hypertext versions of each of five different documents, each sharing a common set of nodes with the four others (with \( p \), the number of nodes, ranging between 23 and 347), but each having a different set of links inserted among the nodes. Each hypertext version of a document was subsequently considered as one of twenty-five complete and separate hypertext databases.

For each database-pair, consistency values were computed using (i) three different types of link-subsets, (ii) three different types of graph-theoretic representation (adjacency, distance and converted distance matrices), (iii) a variety of sets of attribute-values of two types (matrix-element and topological-index), each derived using a different formula, and (iv) twenty-seven different similarity coefficients. As we noted in Section 2, the concern of the present paper is with methodology, and it should be emphasised that it is not our intention for results to be presented here: a summary is provided by Ellis et al. [8], and full details appear in a BLRDD report [7]. Nevertheless, we can briefly record the principal conclusion drawn from our study, which was that levels of inter-linker consistency are comparable to those recorded in studies of inter-indexer consistency, i.e., that they are rarely high, and that they do display marked variation. We proceeded to conduct the second stage of our experiments [9], in which we measured the effectiveness of searches carried out in our hypertext databases by volunteers. Finally, we compared our two sets of measurements, and were unable to reject the null hypothesis that there is no positive association between inter-linker consistency and retrieval effectiveness.

That we were unable to reject our null hypothesis was not too great a surprise given the nature of the evidence that historically has been used to support the claim that there exists a relationship between inter-linker consistency and retrieval effectiveness. Although the number of studies of consistency conducted to date is large, empirical data confirming the existence of such a relationship is notoriously rare; for most authors, the sole source of such data is the study reported in Leonard's unpublished thesis [17]. On inspection of Leonard's results, it remains unclear whether the historically accepted interpretation of their significance is indeed an appropriate one.

5 CONCLUSIONS

If we had found that levels of inter-linker consistency, like those of inter-indexer consistency, were predictive of levels of retrieval effectiveness, then those results would have had ominous implications for the evaluation of retrieval systems that access databases in which hypertext links have been created manually, especially given the intensive nature of the manual labour required in the creation of hypertext links. But we did not find this to be so, and we are therefore not in a position to draw conclusions about the consistency–effectiveness relationship that are equivalent to those drawn by Leonard [18].

However, we believe that we have established a workable methodology for experiments to be undertaken in the future on a much larger scale than that of the experiments we have described in this chapter. Although the methods that we have used in the calculation of similarity values will in their essence be familiar to those experienced in traditional information retrieval research, it is important to note that their application in the current context is almost wholly novel. Although the basic directed-graph model of hypertext is considered axiomatic in much of the published work on hypertext structure, few authors have made use of computational graph theory in consideration of the variations in structure that may be observed amongst different hypertext databases, the sole notable example of recent work in this field being that of Botafogo et al. [4]. Moreover, although we note the wide range of applications of similarity measures, we contend that their application to hypertext structures is a novel one. In this chapter, therefore, we have placed particular emphasis on the description of methodology, in the hope that the techniques we have developed may be of use to researchers in the future.
Our research raises many issues, suitable for further study, that concern the effectiveness of hypertext systems. It would be of interest, for example, to see whether our results were reproducible using (i) original hypertexts that are not merely conversions of existing texts, (ii) hypertexts from professional hypertext authors, (iii) links created automatically by means of statistical analysis of term occurrences, and (iv) expert searchers, inter alia. A particularly fertile area for future research, however, would be a comparison of the differential effect on retrieval effectiveness of link structure and of search strategy. If it were found that the precise shape of a link structure is relatively unimportant in terms of its ultimate influence on retrieval effectiveness, then we would yet be forced to draw a disturbing conclusion: that the presumed objective of a manual linker’s efforts—to construct link-sets on whose account the effectiveness of future searches will somehow be optimised—is largely doomed to failure.

REFERENCES


PART III

INFORMATION RETRIEVAL FROM HYPERTEXT