ENCYCLOPEDIA OF LIBRARY AND INFORMATION SCIENCE

Executive Editor

ALLEN KENT

SCHOOL OF INFORMATION SCIENCES
UNIVERSITY OF PITTSBURGH
PITTSBURGH, PENNSYVANIA

Administrative Editor

CAROLYN M. HALL

ARLINGTON, TEXAS

MARCEL DEKKER, INC. NEW YORK • BASEL

www.dekker.com

It follows that signs of differences $\hat{I}^{i_1}_{10} - \hat{I}^{i_2}_{10}$ and $\tilde{I}^{i_1}_{10} - \tilde{I}^{i_2}_{10}$ coincide, hence the single-valued measure I_{10} has the order preservation property.

Finally, we note that the results considered in this section have so far only theoretical value, because single-valued measures are not applicable in practice without a clear idea of their boundaries of applicability.

REFERENCES

1. F. W. Lancaster, *Information Retrieval Systems: Characteristics, Testing, Evaluation*, Wiley, New York, 1979.
2. C. J. van Rijsbergen, *Information Retrieval*, 2nd ed., Butterworths, London, 1979.
3. G. Salton, *Dynamic Information and Library Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1975.
4. V. S. Cherniavsky and D. G. Lakhuty, *Nauchno-Teknicheskaya Informatsiya* (NTI), 2(1), 24–34 (1970).
5. V. G. Voiskunskii, "Evaluation of Search Results: A New Approach" *JASIS*, **48**, 133–142 (1997).
6. A. N. Kolmogorov and S. V. Fomin, *Elements of the Theory of Functions and Functional Analysis*, 2nd ed., Nauka, Moscow, 1968.
7. F. Gebhardt, *Info. Proc. Mgt.*, **11**, 59–65 (1975).

VLADIMIR G. VOISKUNSKII

INFORMATION RETRIEVAL TECHNIQUES FOR THE AUTOMATIC CONSTRUCTION OF HYPERTEXT

Introduction

The collection of documents that is stored and managed by an information retrieval system (IRS) is usually large and can range from thousands to millions of documents. The collection can be imagined as a set of documents on a flat surface; each document in the collection is linearly composed of textual parts. Together with the collection of documents, the IRS manages an associated indexing terms structure for representing the semantic content of each document, as well as serves to select and retrieve documents judged relevant to a given user's query. The indexing term structure is an access point to the collection since an index term is linked to the associated documents, and the term co-occurrence can support the identification of links among documents and among terms.

These are therefore two main parts managed by an IRS: the collection of documents and the structured indexing terms collection. These two parts can be represented and managed by a hypertext system. In fact, the collection of documents can be managed as nodes of the hypertext, and the network of links can be structured to connect semantically or structurally related documents or fragments, as the structured

The network of links constitutes the main structure that can be used to *navigate* the hypertext. In order to do so, the user needs a tool able to follow the links; this capability is usually provided by a tool called *browser*. A browser usually incorporates both navigation and browsing facilities. If a link does not exist between two nodes that are semantically related, they cannot be viewed (retrieved) by a user who is browsing the hypertext. The only way in which two or more related nodes that have not been explicitly connected by links can be retrieved is by searching the network for some word, string, keyword, or attribute value that nodes have to share. This means making use of an implicit link between nodes. Normally it is only one specific and exact string, keyword, or attribute value that can be used for searching in such circumstances. Few present hypertext browsing tools of stand-alone hypertext systems usually provide more advanced searching functions, such as the exact match retrieval techniques that use a query language based on Boolean logic and that are available in most of the present operational IRS. On the contrary, the World Wide Web (WWW) is a large hypertext in which fairly sophisticated search engines operate, although they are only vaguely incorporated into the browsers themselves.

The capability of easily linking different pieces of information, considered very important in the development of effective HIR applications, can produce IR hypertexts that are very difficult to use by the end user because these same capabilities can generate *user disorientation* and *cognitive overload*. Two main actions that give the opportunity to reduce the risk of user disorientation and cognitive overload are the adoption of a conceptual model as a reference frame to be used in the design and implementation of the hypertext on the one hand, and the evaluation of the hypertext on the other. The aim of the conceptual model is to make the conceptual architecture and then the semantics of nodes and links clear to the final user. If the final user understands what a node or a link means and the way it has been organized by the schema, he can retrieve the relevant information more effectively than is possible without any reference provided by the specific hypertext application. The aim of evaluation is to compare the expected effectiveness to the actual results observed during a laboratory or operational test with or without the direct intervention of real final users. These two steps are further addressed in the subsection "The process of Construction" and the section "Evaluation" below.

CLASSIFICATION OF NODES AND LINKS

A wide range of types of links and nodes is of basic importance in the construction of hypertexts. In fact, the knowledge of a specific type of node or link sometimes makes possible the construction of a structure that would otherwise not be possible to build. This means that knowing what range of types is available in a specific operative environment gives the designer the full range of development and implementation possibilities.

This section introduces two possible classifications of different types of hypertext links and nodes that have been proposed by different authors and in different implementations.

A Classification of Hypertext Links

We would like to draw the reader's attention to the different types of links that can be built in a hypertext. The hypertext can be built by making use of the different characteristics of these different types of links.

In a hypertext, a link implements a logical connection between two related nodes.

- The *origin node:* the node from which the connection starts
- The *destination node:* the node in which the connection ends

Two nodes that have to be explored or viewed sequentially are connected by a *link*. Each sequence of nodes connected by links constitutes a possible exploration path of the hypertext.

Different types of links have been defined, depending on the functions that need to be implemented by the hypertext. Most hypertexts are built making use of the following three types of links:

- *Structural links:* connect nodes of the hypertext that are related by the structure of the document itself. Examples of this type of link are those connecting a chapter with the chapter that follows it or a table of contents with each section reported in it. If the transformation is made from a flat document such as a book to a hypertext, all the chapter/section/subsection structural connections can be rendered in the hypertext using structural links. Each structure (e.g., tree, graph) can be rendered using structural links.
- *Referential links:* are based on some sort of reference the author of the original document has used. A typical link of this sort is the link that implements a reference between the source document and a document that is cited by it.
- *Associative links:* represent undefined associative connections between nodes. They are built making use of content-based connections between fragments of text of the same document or documents of the same collection.

All those links are made explicitly available to the user through the hypertext network. Another type of link, the *aggregate link*, is made available only in few hypertext systems in which the aggregation abstraction mechanism is designed and implemented to give the hypertext designer the possibility of aggregating nodes that together form a new kind of node.

It is possible to add another classification of links, because it is based on a different scope of interest and it can be useful in using the hypertext. This classification categorizes the links as explicit or implicit ones.

- An *explicit link* is one that makes available an explicit reference between two nodes; explicit links are built during the authoring process and they constitute the main part of the hypertext network.
- An *implicit link* is a link that is implicitly present in a node. An implicit link can be activated using a word present in a node. For example, if a user asks to see all nodes that contain a particular word, all nodes containing that word can be made available to the user, and these implicit links are created at run time. This means that an implicit link does not link a pair of nodes, but is implicitly present in the node text and created and made available at run time.

A Classification of Hypertext Nodes

Information retrieval hypertexts can be viewed as document bases, in which a document is an atomic datum connected by links to other documents. These documents are usually unstructured; that is, without any organization into subparts. This means that the concept of node usually corresponds to the one of abstract or bibliographic reference, and the purpose of a node is to store retrievable information. Such a correspondence is due to the use of IR hypertexts as tools to access a classical document collection, in which each document is a bibliographic reference, perhaps together with an abstract.

Classifying nodes according to some criteria is convenient as full-text large documents are made available more and more in machine-readable format. In such a way, each document can correspond to more than one node, and the purposes of a node can vary. The simple classification of nodes we provide in the following is based on the possibility of automatically constructing or detecting them from full-text documents.

In an IR hypertext, there are two main types of nodes.

- *Textual nodes* store text to be read by the final user navigating the hypertext. Textual nodes may be organized according to a more or less structured fashion, may be written in one or more languages or may be of different sizes. Textual nodes conceptually correspond to the usual documents of a classical IR system. If no network of links was made available for browsing, the collection of textual nodes would be equivalent to the usual collection of documents.
- *Index nodes* provide the final user with anchors to reach other nodes, either textual or index ones. They differ from textual nodes because they tend to support the user in his "navigation" among the documents. Index nodes are the ones that make the IR hypertext different from a classical IR system. Index nodes do indeed implement the semantic connections between textual nodes. The semantics of links between textual nodes can be made explicit to the final user by visualizing, for instance, the number of links, the criteria by which anchors have been ranked, and thereupon the weight given to the link.

With regard to the subject of this article, textual nodes are the result of the process of automatic construction of nodes as described in the subsection "Automatic Construction of Nodes" below, whereas index nodes are the result of the process of automatic construction of links as described in the subsection "Automatic Construction of Links." As index nodes implement links, a classification of them would correspond to the classification of links made in the subsection "A Classification of Hypertext Links" above.

- *Structural index nodes* implement structural links. A structural index node is, for instance, a list of anchors linking textual subparts. For example, if a chapter/section/subsection organization takes place, structural links are based on the corresponding headings.
- *Referential index nodes* implement referential links. A referential index node is, for instance, a list of anchors linking referred documents. For example, such an index node can be a list of bibliographic citations to a reference list.
- *Associative index nodes* implement associative links. An associative index node is, for instance, a list of anchors linking semantically related documents. For example, such an index node can be a list of anchors to similar documents, where similarity relationships can be detected at browsing time or at the time of authoring.

- *Explicit index nodes* implement explicit links. An explicit index node is, for instance, a structural or referential index node, since such information can be detected at authoring time.
- *Implicit index nodes* implement implicit links. An implicit index node is, for instance, an associative index node storing anchors to similar documents, since the latter can be dynamically detected at browsing time and updated over time as new documents are inserted into the database.

We classify textual nodes according to the following criteria:

1. *Structure:* Text can be either *structured* or *unstructured*. In the former, an organization in parts and subparts, such as sections, paragraphs, or sentences, can be detected and made available to the process of hypertext construction. In the latter, no organization exists or can be detected, as is the case, for example, with plain text "windows" opened on the document. The possibility of using or detecting text structure helps construct nodes that are more or less organized in the same way as the full-text document. In some applications, text structure can be crucial in the IR hypertext effectiveness since the chapter/section/subsection structural connections, for instance, are made by the document's author through an intellectual authoring work.
2. *Content:* Node contents can be either *homogeneous* or *heterogeneous*. The distinction between homogeneous and heterogeneous contents is important since the effectiveness of the IR techniques employed in the automatic hypertext construction, the subject of this article, depends on the content heterogeneity of the full-text document. Word ambiguity, one of the facets of heterogeneity, makes IR techniques less effective because they are often based on statistical data given by the co-occurrence of keywords in documents. Since the degree of word ambiguity is directly related to the extent to which the document content is heterogeneous, the effectiveness of IR techniques-based automatic hypertext construction methods depends on the degree of word ambiguity.
3. *Size:* The issue of node size is important from two different points of view.
   a. *User's point of view:* IR hypertext are interactive tools presented to the final user through screen-based interfaces. Leaving aside the issues related to electronic hypertext publishing, the size of a node can influence the effectiveness of the interaction between the final user and IR hypertext. The size has to be carefully chosen according to the purpose of nodes, the hardware and software features of the interface, the type of final user, and other related aspects. If the textual node content has a presentation purpose, the node should be large enough to include all necessary data to illustrate the topic. Nodes that are too small, however, can be poorly informative due to the scarcity of data contained in them. If the node works as an anchor, it should be small, since the user is likely to select it to carry navigation rather than to carefully read it. Nodes that are too large, however, can be ineffective, since the user will be unable to read them through a screen.
   b. *System's point of view:* This is the view most relevant to the subject addressed in this article since the core of the automatic hypertext construction is automatically performed by a system. As textual node contents are provided as machine-readable documents stored in files, the automatic node extraction from a variable number of files of different sizes performed by the system is important. The resulting nodes are to be of adequate size since nodes that are too large are likely to have a heterogeneous content, thus making the employed IR techniques ineffective. Attention should be paid as well to nodes that are not so large, since they can have a heterogeneous content if they address different topics in a few short sentences. The availability of a text structure that can support the system in detecting the candidate nodes can make the extraction of small nodes from a large text easier.
4. *Language:* One aspect of the document collection heterogeneity is the diversity of languages. By *multilingual IR* (MLIR) we mean the task of retrieving documents written

in one or more languages to answer a query expressed either in one of those languages or in a different one. The complexity of MLIR, especially in Europe, is due to the variety of languages that can occur at the level of an individual document, a whole document collection, or an individual user's query. The main limitations of current IR techniques is that much of the IR work has been done for English-language text documents only and that work cannot be superficially exported to other language domains. It is sufficient to think of the problems of word ambiguity and of multiword expressions. The problems related to multilingual collections are to be addressed by the researchers working in the area of automatic hypertext construction if multilingual document collections are to be transformed into MLIR hypertext.

AUTOMATIC CONSTRUCTION OF HYPERTEXT

Hypertext for IR is considered as at least a potentially effective alternative model to the classical Boolean, vector space or probabilistic model, since the way of interconnecting nodes through links and the browsing-based search strategies seems naturally close to the classification, indexing, and searching process carried out for decades by indexers or librarians, for instance.

Any IR hypertext can be built by its author or groups of authors from scratch. Nowadays, however, the most common situation is the construction of a hypertext starting from a collection of "flat" documents available in a digitized form due to the above-mentioned availability of textual data.

The growing size of document collections, and specifically the wide availability of Web pages makes IR hypertext manual construction impossible to achieve for two reasons.

- The high number of nodes and links to be built makes manual hypertext construction infeasible.
- The individual full-text documents currently available in machine-readable format are too large to be used as nodes directly.

To address the problem of size and number of documents stored in the current collections, methods and tools have been proposed to automatically detect links and to extract from the document smaller fragments to be used as nodes and to automatically update the hypertext by inserting, modifying, and canceling part of it over time.

In particular, most of the proposed methods for automatic hypertext construction are based on IR techniques since researchers in IR have always dealt with the automatic representation of semantic relationships between IR objects (i.e., documents and terms).

Information retrieval techniques can in principle be employed to automatically build hypertexts other than for IR, and an IR hypertext can be automatically constructed using non-IR techniques. The focus of this article is on hypertexts automatically constructed for IR, however, and therefore the task we consider is that of a user wishing to retrieve relevant documents from a large document collection. More specifically, we address the use of IR techniques for automatic hypertext construction, and we leave aside the use of non-IR techniques. The reasons why much research has been done to apply IR techniques for the automatic construction of IR hypertext can be summarized in the following:

1. Hypertexts are basically networks of links expressing semantic relationships, and as was already stressed above, researchers in IR have dealt with semantic relationships for decades since IR techniques are designed to disclose such relationships.
2. Automatic hypertext construction has on the one hand become necessary to address the problem of large documents or collections. On the other hand, IR techniques are designed and implemented to deal with large documents or collections.
3. The actual document collections are provided to the final users by means of IR systems for IR purposes. A hypertext that is automatically constructed starting from these collections is therefore the main candidate for being a tool for IR purposes as well.

One can think of employing non-IR techniques to automatically construct IR hypertexts. It is therefore important that such techniques are able to disclose the semantic relationships necessary to support the final user during the IR tasks. Conversely, it is possible to employ IR techniques to automatically build hypertexts specifically designed for non-IR tasks, such as learning, teaching, data finding, or exploration. It is therefore important that IR techniques are tailored and adapted to these specific application domains. In fact, current IR techniques are designed for search tasks based on queries with a low degree of interaction between the user and the IR system. Investigations have to be made to test whether they perform well for highly interactive tasks, such as learning or teaching performed by the final user of a hypertext system.

THE PROCESS OF CONSTRUCTION

Building the hypertext requires the following main steps:

1. *User's requirements analysis:* identification and design of the target hypertext application; that is, what kind of hypertext application the author wants to produce given the target final user community. The output of this step consists of the description of the target final user community, the definition of the specific tasks to be performed, and the list of users' requirements about the types of nodes and links, as well as the functions the IR hypertext should provide.
2. *Design:* formalization by means of a model of the IR hypertext conceptual schema and of the retrieval functions. The hypertext schema is the conceptual reference for the tool that automatically builds the hypertext. The retrieval functions are the operations the user can perform through the hypertext to retrieve relevant information.
3. *Construction:* transformation of the initial individual big document or collection of documents into a hypertext. The process requires the initial input to be processed to extract the nodes, and links between fragments to be built between the nodes. Automatic construction is based on techniques that can be automated by means of programs running on computers. As IR techniques have been implemented by computers since the early days of IR, they are the natural candidates for being the basic algorithms for automatic hypertext construction.
4. *Visualization:* making the hypertext available to the potential user community. To reach this final aim it is necessary to have a presentation and browsing tool that makes human–computer interaction (HCI) capabilities available to final users.
5. *Evaluation:* the resulting IR hypertext has to be tested by the target user community to gain insights about its effectiveness in retrieving information. The evaluation results can be fed back to the design or construction steps to get an improved version of the hypertext.

This article concentrates on automatic hypertext construction, and therefore we will ignore some of the other steps of the whole process: the user's requirements analysis, design, and visualization. We address evaluation in the section "Evaluation" below. We will, however, stress some aspects about the user's requirements analysis and visualization that are related to the automatic construction of an IR hypertext.

A hypertext can be automatically built for different purposes according to the analyzed user's requirements. Other than IR, learning, teaching, data finding, referencing, or exploration are examples of tasks the hypertext can be required to support. By referencing we mean a user wishing to retrieve information that is useful to complete his knowledge. For example, a textbook can be automatically converted into its hypertextual version to be used for reference, in particular after the subject dealt with by the textbook has been already studied. This is particularly true of scientific textbooks, with formulas and data that cannot be easily memorized by the reader but that are known to the reader and therefore can be recalled quickly once accessed in the textbook. The same hyper-textbook can be used as a tool for studying a subject to learn that subject without being supported by any paper-based textbook or lecture given by the teacher. "Data finding" is the process of locating a specific record, for instance, a bibliographic record for which the user wishes to know the value. Moreover, the final user can navigate the hypertext to learn or refer something, but in a way that he reaches other sources of information that allow him to refine his own information need. The latter is called "exploration" and differs from retrieval since in the former the information need is only partially formulated and can be dynamically refined during browsing, whereas in the latter the need is almost completely formulated and is clear to the user who formulates it into, for instance, a query.

With regard to *hypertext visualization*, it is necessary to stress that the problem is twofold.

1. *Visualization technology:* In principle, choosing the type of technology to visualize the results of the automatic construction of a hypertext is necessary. In reality most applications tend to use WWW technology (*2*) as presentation technology. This means that the functions and capabilities for visualization that are available through Web technology need to be taken into consideration and used. From the application designer point of view, this means that using Web gateways or other tools becomes necessary to exchange data between the user and the data (links and nodes) storage system. Different ways of solving the querying of a data management system through the Web are presented in Mendelzon et al. (*3*) and in Agosti et al. (*4*). The visualization capabilities that are not supported through the Web cannot be used by the designer; one example of the types of capability that we have in mind that are not supported are those presented in Hearst and Karadi (*5*).
2. *Visualization methods:* The visualization of a hypertext that has been automatically constructed is not a straightforward process. The designer of the automatically built hypertext needs to choose what visualization paradigm to adopt and what methods and tools to use for the visualization of the hypertext to the user. This problem has been addressed in Salton et al. (*6*), in which the authors propose and use a method that uses both full-text documents and text passages and computes similarity measurements between full texts and passages generating "text relation maps" to present the hypertext to the user. Text similarities are shown through these maps.

SUGGESTED BIBLIOGRAPHY

Some of the initial papers addressing the issues related to the combination of hypertext and IR capabilities to produce a new sort of innovative information management tool are Refs. *7–9*. The collection of papers in Agosti (*10*) presents the research results at the time of publication of that special issue, and in Agosti and Smeaton (*11*), the effort has been made to present the merging of the two areas of IR and hypertext in an integrated way.

The aggregate link mechanism is based on the "database aggregation" abstraction mechanism that was initially introduced in the database area and presented in Smith and Smith (*12*). This mechanism was initially made available in semantic data models (*13*), but is still rare in operative database management and hypertext systems.

An example of the process of detection of implicit links is described in Aalbersberg (*14*).

In Belkin et al. (*15*) an analysis is presented about the types of tasks the final user can perform.

A case study on the automatic construction of a hyper-textbook for learning, teaching, or referencing is given in Crestani and Melucci (*16*).

The papers collected in Bieber and Isakowitz (*17*) provide a comprehensive survey of the most important results of hypermedia design and manual, or almost manual construction. To name but a few, the hypertext data model (HDM; *26*) and the Dexter model (*27*) are among the most important hypermedia models. The emphasis of these papers is more on the design and evaluation phase than on the construction phase, so no automatic construction method is required or addressed. From the HIR point of view, the contributions reported in Agosti et al. (*18*), and specifically Lucarella and Zanzi (*28*) and Chiaramella and Kheirbek (*29*) are in general relevant to the area, and in particular they are important for a deep understanding of the importance, motivations, and methods underlying the IR hypertext design and modeling.

References addressing automatic construction are referred to in the following sections of this article. Meanwhile, we can organize the most relevant references to research work by classifying them according to the task and the types of techniques used for the automatic construction process. In Table 1, each column corresponds to a technique type (i.e., IR or non-IR technique), and each row corresponds to the task (i.e., IR or non-IR task). Each cell of the table includes the references to research

TABLE 1

Techniques and tasks by Reference number

	IR techniques	Non-IR techniques
IR task	*18*	*22*
	19	*23*
	20	*20*
	21	
Non-IR task	*16*	*24*
	22	*25*

work that mainly employed one of these techniques to perform one of these types of tasks.

Research in MLIR is currently at the beginning of its history, and therefore the most interesting and relevant research works are still at "prototype" level. We would suggest, among the others, Refs. *30–35*.

The Use of Information Retrieval Techniques for the Automatic Construction of Hypertext

INFORMATION RETRIEVAL ACTIVITIES

Information retrieval is a general term referring to the retrieval of multimedia data from large unstructured collections of items called documents representing an information content relevant to a user's information need. The most consolidated results are in the domain of textual documents, and therefore the term information retrieval is often used as synonymous with "document retrieval," or a little more generally, with "text retrieval." In this article, we adopt such synonymy since we concentrate on hypertexts automatically constructed from textual material.

An IRS performs two main activities: indexing and searching. Indexing is the activity by which the system or a human indexer (perhaps supported by the system) manually, semi automatically, or fully automatically describes documents and queries (i.e., representations of the user's information need). Searching is the activity by which the system supports the user in retrieving the relevant documents.

Indexing is the basic activity performed by a system that automatically constructs a hypertext on the basis of IR techniques. This is because IR techniques are based on document descriptions resulting from the indexing activity. The effectiveness and the mechanism of automatic hypertext construction algorithms therefore heavily rely on the effectiveness and the mechanism of indexing algorithms.

Searching traditionally starts with a query submitted by the user to the system to produce a list of documents that are perhaps ranked by a measure expressing the degree to which the system believes that a document is relevant to the query. Research work in HIR has addressed the problem of integrating query-based searching to browsing or navigation, which are the main activities performed with a hypertext system. The tools for the automatic hypertext construction should therefore provide the final user with the possibility of using both query-based searching and browsing in an integrated fashion.

The growing availability of large, full-text, and heterogeneous documents in machine-readable format has been observed in recent years. Large texts include reports, complete papers, book chapters, books, or encyclopedias. The meaning of large depends on different variables that can be summarized in the available computational resources needed to manage such a quantity of data and in the capabilities provided by a computer to make large documents usable and useful for the user. The less the power of computational resources or the worse the system capabilities of providing the user with usable and useful access to large documents, the more the

documents are to be considered as large. Information retrieval techniques have been designed to manage documents as an individual and atomic unit, perhaps homogeneous in content, such as bibliographic records or short abstracts. Classical IR techniques are thus inadequate to manage large or short but heterogeneous documents, since IR algorithms are unable to deal with, for example, word ambiguity, which is typically present in large and/or heterogeneous documents.

Automatic hypertext construction methods based on IR techniques inherit the difficulty in managing large documents. To make the use of large or heterogeneous texts more effective, we need techniques and tools to extract short texts from larger ones to be matched against the user's information needs, to be linked through relationships, or to be assembled together to generate homogeneous texts that can be managed by classical IR techniques as well. A number of advanced techniques are being developed to address the problem of extracting information from large or heterogeneous documents. To name but a few, passage retrieval, automatic abstracting and summarizing, text structuring, and theme extraction are research areas dealing with highly structured documents, variable-size documents, or heterogeneous topic documents.

INFORMATION RETRIEVAL: BASIC TECHNIQUES

This section presents the basic techniques of automatic IR that are relevant to automatic hypertext construction. These techniques are automatically performed by the system as support to the user, who is not required to be able to perform them. These IR techniques are those performed by the IR engine component of the HIR system and are used by the IR engine in a way that remains completely transparent to the final user, who is not aware of their use, and they are not presented to him through the system visualization interface.

- *Stop-words elimination:* A list of stop or fluff words is used to remove from documents and queries the commonly occurring words that are of little use in retrieval. The input is the so-called stop list and the document or query. The output is the document or the query without the words contained in the stop list.
- *Conflation:* Words of documents or queries are mapped to a class of single forms by removing all the variants. The most used conflation algorithm is called "stemming," in which each word is reduced to its root by removing its derivational and inflectional affixes. The input is then a document or a query, and the output is the document or the query. Both are described as a set of conflated words, perhaps without stop words.
- *Term or phrase construction:* Using lexicon, dictionaries, and grammars the words can be related together to form multiword term or phrases. An index term can therefore be either an individual word or a more complex multiword term or phrase.
- *Weighting:* Each document or query word is assigned a numerical score, called "weight," expressing its representational power within the document or the query. The most important weighting scheme is the $tf \cdot idf$ scheme based on the product of two functions: the first directly proportional to the frequency of the word within the document or the query, the second inversely related to the number of documents within which the word occurs. The basic $tf \cdot idf$ scheme formula is like $tf_{dt} \times \log \frac{N}{n_t}$ where tf_{dt} is the frequency of occurrence of term t in document d, N is the total number of documents, and n_t is the number of documents indexed by t.

The importance of weighting in searching activities is due to its discriminating power between relevant and nonrelevant documents as tested and proven by several experiments. Weighting schemes are fundamental to the effectiveness of searching activities, of the advanced techniques described below, and therefore of the automatic hypertext constructions that are based on them.

- *Similarity computation:* The system decides whether two objects (i.e., documents or queries) express a similar information content to each other by computing a numerical score, simply called "similarity." For example, if the vector-space model is employed, the similarity is the cosine of the angle between two vectors, each describing an object. If the probabilistic model is used instead, the similarity is the value of a function directly related to the probability of relevance of a document with respect to either a query or a document. A threshold can be set to decide whether two objects are likely to have a similar information content. If the similarity value computed on the object description is over the stated threshold, the information content of the objects themselves are judged as similar.

  Similarity can be computed between two objects, each taken as a whole, or between parts of objects. The technique is used with two different scopes.

  - If similarity is computed between two objects, each taken as a whole, then the technique can be used to produce a global measure of similarity between two documents or between a document and a query (*global similarity*).
  - If similarity is computed between pairs of parts of objects, then the technique can be used to produce a similarity between objects that is based on their maximum similarity between parts (*local similarity*).

    One can show that there is a strict correlation between global and local similarities, and that local similarity is more precision-oriented than global similarity and can be used to refine the retrieval result of the latter.

INFORMATION RETRIEVAL ADVANCED TECHNIQUES

The previous section was devoted to the presentation of the IR techniques that are relevant in general to automatic hypertext construction. In the IR area some other more advanced techniques have been developed and can be of use in the automatic construction of a hypertext that supports advanced IR capabilities. These advanced techniques are presented in this section.

- *Text clustering* is the task of grouping together textual documents according to a given criterion. A possible criterion is that of *similarity*, which can be employed to calculate similarities among documents; for example, if the vector-space model is employed to describe the document semantics, measures of similarity among documents can be computed using the cosine function. The aim of clustering is both to reduce the number of objects that need to be managed in a real application, and to improve the retrieval effectiveness. Clusters can represent groups of documents, therefore clusters can be managed as surrogates of documents. The number of clusters is smaller than the number of documents; therefore the system can manage clusters more efficiently than documents. In this way the efficiency of the retrieval system is improved. The cluster hypothesis states that relevant documents tend to be similar, and then once clustered together via similarity measures they tend to improve the retrieval effectiveness of a system.
- *Text structuring* is the task of decomposing a large input text into smaller pieces according to a given external scheme or to the one employed by the text's author to write the text itself. The output of structuring can therefore be twofold: the text structure and the component subparts. The aim of text structuring is to provide the user or the system with a text that is more organized and then more easily manageable than an unstructured one. We recall that in fact both organization and titles of parts carry much useful information.
- *Theme extraction* is the task of extracting from a large input text the topics addressed within

the text itself by the author. The output of this task isthen a list of multiword terms representing the extracted themes, between which automatic links can be inserted. The purpose of theme extraction is to provide the user or the system with an idea about the different subjects treated by a heterogeneous large documents. The user is then supported in selecting the relevant information, and the system can take advantage of such an organization during the retrieval of relevant documents and the rejection of the irrelevant one.

- *Summarization* is the task of creating a text from one or more larger texts. The output is then an individual text that includes the most important topics. The summary is in a form that makes it usable in further tasks, such as indexing and retrieval, other than the ones we are describing. Like theme extraction, summarization would provide an idea about the subjects addressed by the input texts. Different from themes, summaries have to be "well-formed" to be used as "stand-alone" documents.
- *Passage retrieval* is the task of identifying and extracting fragments from large (or short but heterogeneous) full-text documents. Passages are chunks of contiguous text belonging to a larger text. What makes passage retrieval different from the previous tasks is that passages are necessarily matched against a query, they can serve as devices to retrieve larger documents, or they are extracted after large documents are retrieved.

AUTOMATIC CONSTRUCTION OF NODES

The automatic construction of the nodes to be used in the final IR hypertext is a mandatory step if the initial input collection is made of documents that cannot be directly employed as nodes. Documents can be indeed too large or too small, can have a heterogeneous content or be written in different languages, or can have a structure that may be useful to consider to extract nodes or to detect relationships between nodes.

Methods for the automatic construction of nodes can be based on one of the following:

- The trivial one that is based on taking an individual document as an individual node
- Text structure to identify parts of a larger documents to be further used as nodes
- Text content to create new nodes starting from other ones

The trivial method to create nodes is the best one whenever input text is already organized into different documents of adequate size. If *text structure* is available and if its use makes sense, a tool for the automatic construction of hypertexts has to be able to detect the different parts composing the whole input textual document. The detected parts can be used to be further processed or as hypertext nodes between which links can be built. The techniques used to detect the structure are straightforwardly based on parsing algorithms given a "syntax" by which the documents are written. Some examples of these are HTML, LATEX, or other SGTML-based mark-up languages. Extracted document parts, if available, can be

- Catalogue data, such as the ones managed by a system for the retrieval of bibliographic data, including title, author's name, and date of publication. These catalogue data can be treated as nodes as a whole without fragmenting them into the individual fields.
- Logical sections that the document author has organized within a hierarchical structure to represent the subject addressed in the document. Some examples are the abstract, sections and subsections, footnotes, lists, bibliographic references, and subject indexes.

The initial input textual documents or the parts detected from them can be processed on the basis of their own *text content* to produce an additional set of nodes. Methods based on text content aim to

- Remove the content heterogeneity
- Deal with multilingual documents or document parts
- Arrange for adequate node size
- Give a structure to unstructured documents or parts

The advanced IR techniques that can be employed to automatically build nodes from documents or document parts using the text content are

- *Text clustering*, which can be used to produce a node as the result of a combination of the full text of clustered nodes. The result can give an idea to the final user of the content of the clustered nodes in terms either of their full texts or of their descriptions, such as keywords. If information about the component nodes is maintained, the links from the cluster to the latter can be built, and the cluster can be used to reach the nodes by browsing these links. Text clustering can help reduce the size of large documents and group together the documents dealing with a homogeneous content. The effectiveness of clustering in automatic hypertext construction can be explained by the cluster hypothesis, which states that document nodes that are relevant to the same query tend to be linked together.
- *Theme extraction or summarization*, which can be used to produce nodes storing the main themes addressed in the input documents. The theme nodes can then be used by the system to shortly communicate the document subjects to the final user, who can decide whether or not the documents are of interest and whether or not to navigate the proposed links. This technique can be of help to give the user an idea of the degree of content heterogeneity or of multilinguality. The user can then decide the specific theme or language to select and follow the appropriate link.
- *Text structuring*, which can be used to extract parts from hierarchically organized large documents. The resulting parts are smaller and of more homogeneous content than the initial document. Since text structuring deals directly with document text, the result are nodes with their own textual content that can be further processed to, for example, compute links or cluster into new nodes. Text structuring requires that an adequate structure is available with the input document, but once it is available this technique can support the creation of nodes of adequate size, unless the parts are too large to make summarization or theme extraction necessary.
- *Passage retrieval*, which is useful to identify text fragments to be used as nodes and to build content-based links between the passages. Passages could also be used as anchors in addition to nodes, since passages can be adequately small, such as a sentence or two or three words, and maintained inside the text of a node. The techniques can help deal with large documents whenever they cannot be easily decomposed into smaller parts, or whenever one has to build links between a few large documents, instead of many small ones. The resulting hypertext can be made of few nodes, but links are inserted between passages that work as anchors.

Methods for the automatic construction of nodes based on text content aim to change the form by which documents express their own informative content since the changed form is more adequate within the process of automatic hypertext construction. Since the document informative content is described through indexing rather than full text the text content-based methods for automatic node construction are based on indexing. We will see that an analogous concept states for automatic link

construction, since indexing allows for extracting those semantic content descriptors that can be used to automatically build content-based links.

AUTOMATIC CONSTRUCTION OF LINKS

The automatic construction of links is the core step of the whole process of hypertext construction, since it rapidly transforms a flat document collection into a powerful interactive retrieval tool. Methods for the automatic construction of links can be based on one of the following:

- Text structure to detect explicit and structural links
- Text content to detect implicit and associative links

Methods based on *text structure* can make use of different sources of evidence, depending on the nature of the initial input document collection. These sources are structurally elements, such as outline headings, citations, cross-references, and indices. The resulting hypertext is made of nodes corresponding to the document parts connected by means of structural links. These methods are based on the reasonable assumption that there is a close relationship between the physical components of a document and the hypertext nodes. From an IR point of view, such structure-based hypertextual organization should provide a better understanding of the semantic content of documents. This type of method is well suited for medium-grained documents that are regularly and consistently structured. Regularly and consistently structured documents are those having a well-identified and fixed structure, such as collections of dissertation abstracts, bibliographic cards, and manual pages. Larger or less regular and consistent documents—for example, scientific papers—would require some manual intervention to catch content-based links; that is, links nonexplicitly inserted in the documents meant to represent semantic "aboutness." These latter links are the most interesting from an IR viewpoint, since we are mainly interested in semantic navigation and browsing of the document collection.

There is another facet of the concept of structure in an IR hypertext that can be worth considering: the structure of a hypertext as a graph, where links are the edges connecting the nodes. The graph can be decomposed into subgraphs called "aggregates." Identifying aggregates within existing hypertexts can be useful for analyzing the overall hypertext structure, in addition to using them as a kind of "macro" hypertext node. The final aim is to measure the risk of "user disorientation." This task is mostly done by analyzing link pattern; that is, how nodes are interconnected. User disorientation comes from the high number of links we need to follow to reach interesting nodes. A high number of links sometimes indicates a too complex hypertext structure. The hierarchical structure is often the most natural organization of information. A good rule is that of starting to create hypertexts in a hierarchical way, even though this guideline is often lost because of the intrinsic network nature of cross-referenced documents.

Specific hypertext substructures, such as clusters or hierarchies, and metrics called "compactness" and "stratum" can be defined to help authors in writing hypertexts, and then to solve the user disorientation problem. Hierarchies are important because

the root is a node that can reach all the other nodes with a low number of links. Clusters are sets of related nodes that are identified by analyzing the hypertext structures. The compactness measure helps the hypertext author to assess the density of linking (i.e., roughly speaking, the mean number of links per node). The stratum measure gives an account about the number of links to be traversed to reach a node starting from another node. By finding the roots or clusters of a hypertext or by adopting some metrics, authors can have a more clear idea about structure and then they can indirectly evaluate the effectiveness of the hypertext. Hypertext structures and related measures can be employed in automatic construction processes. Algorithms used to find hierarchies or clusters of hypertexts do not scale up well when very large hypertexts are analyzed, however. Anyway, if one assumes performing hierarchy or cluster extraction in an incremental manner, the computational problem can be overcome by using some modified algorithm. For example, a modified version of the breadth-first-search algorithm for hierarchy identification can be performed only when new documents are authored.

Methods for the automatic construction of links based on *text content* try to catch the similarity relationship between the nodes of the hypertext to be built. Similarity is one of the semantic relationships that occurs between nodes. The most used IR technique to assess whether or not an object's semantic content is similar to another," is similarity computation. Since links between nodes express a semantic relationship between nodes, similarity computation can be used to automatically build this type of link. Similarity computation is performed on *descriptions* rather than on full text, therefore the text content-based methods for automatic link construction are based on indexing. Since the same holds for automatic node construction, almost all the methodologies for automatic hypertext construction based on IR techniques hinge on indexing.

In its simplest form, the algorithm for similarity computation can be formulated as follows, provided a model, the similarity function used to compute similarity values, and a threshold value:

1. Compute the description of two textual node's contents using the model.
2. Compute the similarity score between the descriptions using the function.
3. If the score is over the given threshold, then insert the link between the nodes.

Starting from that simplest form, methodologies that are much more complex and effective for the automatic construction of links can be defined. In the following, we stress the "schools" (universities and research institutions) at which these methodologies were developed and where research is still going on. We note the authorship of each approach since there are no consolidated and unique methodologies, yet one can combine all of them to set up one's own methodology according to the specific requirements and application domain. In the bibliography below we refer to the most relevant research papers written by each school. We illustrate three possible methodologies together with the schools that proposed them.

1. The combination of clustering, local and global similarity (Cornell University)
2. The automatic detection of links and of different link types (Cornell University and the University of Massachusetts)

3. The automatic construction of links of different types between different types of nodes: (University of Padova, Italy)

The Combination of Clustering, Local and Global Similarity

The methodology can be used to create links between text segments to practically build up a hypertext at retrieval time. The technique proposed here is an attempt to use vector similarity to produce a network of text segments that are semantically related.

The basic idea is to use the normalized $tf \cdot idf$ weighting scheme subsection "Information Retrieval: Basic Techniques" above in the context of the vector space model to evaluate the similarity between two text segments.

The same technique can be used to

1. Retrieve text segments in response to a query that can be used in an iterative way to link text segments at retrieval time
2. Link text segments to build a graph where nodes are implemented by the segments and edges are the automatically built links, the similarity value between couples of text parts being the link weight
3. Combine retrieval and automatic construction of a link within a unique framework for querying and browsing an IR hypertext

We describe the three techniques in more detail in the following:

1. *Retrieve text segments*. The first technique works in the following manner:
   a. Retrieve, in response to a query, a set of m text segments using global similarity.
   b. Refine the retrieved set by rejecting all but k text segments. This can also be done in two ways: by setting the value of k first or by employing a local similarity threshold and accepting the k segments that are over that threshold. In the second case the value of k cannot be controlled directly.
   c. Use the retrieved set of k segments as a new set of queries and for each of them restart the process. The process can be repeated n times. At each iteration link the k segments (queries) with each of the k new text segments accepted in response to each query.
   The process produces a tree in which nodes can reappear at different levels and whose depth and breadth can be controlled by carefully choosing respectively m, n, and k.
   The major advantage of this approach is that it can be performed at retrieval time. This could appear as a disadvantage since it could take a long time to produce a large number of links, but links can then be stored and used subsequently when one of these $n \cdot k$ text segments is retrieved in response to a new query. The process does not take into consideration the problem of updating the link structure once new documents are added to the collection, however. Moreover, it is not clear if the text segments' traversal can only be performed going down from the root to the leaves of the tree or in the opposite direction. Also, the problem of identifying the correct dimension of the text segments is left unsolved.
2. *Link text segments to build a graph*. A graph representation that very much resembles a hypertext is introduced. Nodes of this graph can be documents or textual fragments (paragraphs, sentences) extracted from documents. When nodes represent documents, a global measure of similarity is used to measure closeness among documents. This is based on the inner product of the weighted vectors representing the documents in the vector space model. In this case, term weights are computed using the normalized $tf \cdot idf$

weighing scheme. Similarly, when nodes represent text segments, the same inner product is used to measure their similarity. In this case, however, nonnormalized term weights are used, to give preference to longer matching sentences that are more indicative of coincidences in text meaning. An accurate analysis of the structure of a document can be obtained by putting the nodes representing text fragments along a circle and drawing a line whenever the similarity of two text fragments is over a predetermined threshold. Using this technique is possible to decompose documents by identifying homogeneous parts (sets of text segments) of the document. The same technique could be used to link parts of the document that have strong relationships between them. Local similarity is proposed as a precision filter used to discard documents that may have a high global similarity with the query due to language ambiguities that have a low local similarity with the query.

3. *Combine retrieval and automatic construction of links*. The technique is an extension of the vector space model to include a precision-oriented device (a local matching), and it is the same as that already presented in the previous two points, but is here addressed with the dual purpose of retrieval and text structuring. It is this second use that is most interesting to us, since this technique can be used for the automatic construction of a hypertext at retrieval time. The use of the global–local matching technique already explained assures that the links are semantically oriented. Researchers at Cornell University have reported a convenient classification of the different techniques that can be used for automatic text structuring.
   a. "Breadth k–depth n" search: fixed number of documents accepted in response to a query and fixed number of iterative searches.
   b. Decreasing levels of similarity: the number of accepted documents is variable since it is determined by a similarity threshold. The threshold is then progressively increased to produce a self-contained map.
   c. Clustering: performed on the results of a search and incorporated into the final ranking.

The Automatic Detection of Links and of Different Link Types

The main aim of this technique is to automatically describe the nature of the link—that is, to assign a type to each link—instead of constructing the link itself. This is done for arbitrary collections of unrestricted subjects of any size. One assumes that one has an existing flat collection of flat documents and a classification of link types to be automatically detected.

- *Revision* links are those representing versions of the same text segment.
- *Summary* links are those connecting a set of text segments to the one that summarizes them. Summary links are quite different from summarization, the former being already existing text segments that may or may not be a summary of each other; the latter is a process to build text segments that summarize other segments.
- *Expansion* is the inverse of the summary link type.
- *Equivalence* links connect segments that are about the same subject or very close subjects.
- *Comparison* links connect fairly close contents of text segments.
- *Contrast* is the inverse of the comparison link type.
- *Tangent* links connect segments that are marginally relevant to a segment.
- *Aggregate* links are between-text segments grouped together to form a new segment.

The output is a type label for each link and a graph representation of the resulting hypertext. The graph representation helps the final user understand the nature of the links. The technique to detect link types is based on the combination of clustering local

and global similarities as described in the previous section. What makes this technique different is the algorithm for automatically typing links. Let us consider two large documents from which parts can be automatically extracted using the text structure. The algorithm performed on the documents consists of the following steps:

1. Compute global and local similarities between documents, between documents and parts, and between parts.
2. Combine global and local similarity values to classify similarities as different degrees of strength, such as strong, good, and weak.
3. Collapse strong links and merge linked parts to obtain a simplified set of links and parts.

The resulting set of links and parts is a hypertext whose nodes are document parts or an aggregation of document parts. The links between the nodes are the result of collapsing strong similarity-based links. The nodes therefore consist of merged parts, and the links consist of collapsed strong similarity-based links. Let us describe link collapse with an example. Let us consider two documents x and y, and two parts A and B of each of them: (A_x, B_x) and (A_y, B_y) respectively, such that B sequentially follows A. If there are two strong links between the As and between the Bs, and if A is physically near B within each document, then A and B are merged to create a new aggregate part, and a collapsed link is created between the two aggregate parts.

Link types are assigned by analyzing the pattern of the merged parts and the collapsed links. Type labels are assigned to the collapsed links of the resulting hypertext according to the following rules:

- A link is of the *revision* type if the two documents organize their own subjects in the same order and if corresponding parts are connected through strong similarity-based links. For example, the collapsed link between the above-mentioned documents z and y is a revision link.
- A link is of the *summary/expansion* type if the quality of unlinked text of a document is higher than the quantity of unlinked text of the other document; the link from the former to the latter is of the summary type, the reverse one is of the expansion type.
- A link of the *equivalence* or *comparison* type if it is neither a revision nor a summary/expansion link; the choice between "equivalence" and "comparison" depends on the degree of strength of merged links (i.e., strong and good links).
- A link is of the *contrast* type if the quantity of unlinked text of both documents is significantly high.
- A link is of the *tangent* type if few links start from or end at a node.
- A link is of the *aggregate* type if it is between two nodes consisting of parts that form a cluster. (See above about "hypertext structure" and related measure such as "compactness" and "stratum.")

The Automatic Construction of Links of Different Types Between Different Types of Nodes

This sector presents an approach to the automatic construction of hypertexts that has been developed in the context of the research activities of the information management systems (IMS) group of the Department of Electronics and Informatics of Padua University, Padua, Italy. This presentation aims to give the reader an idea of the activities that are necessary to automatically produce a hypertext starting from a

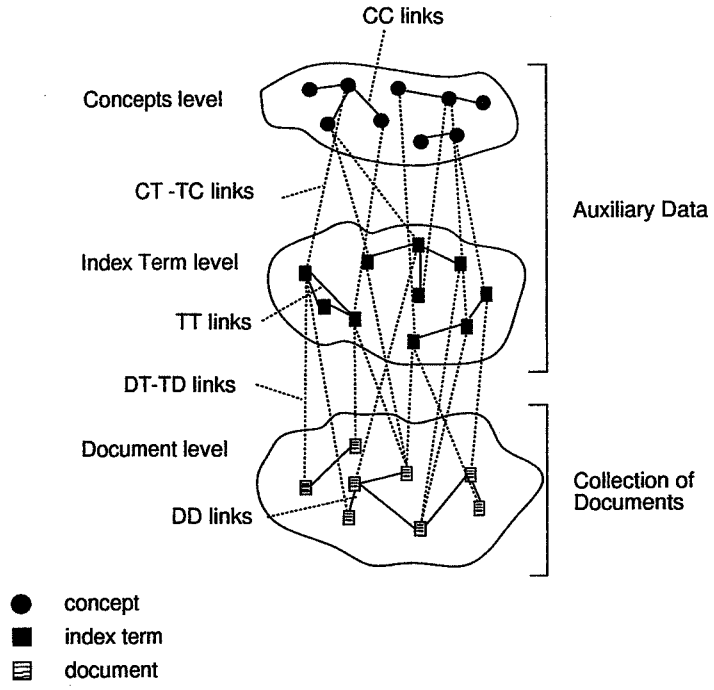

FIGURE 1. *The conceptual architecture of hypertext.*

flat collection of textual documents. The full underlying methods and techniques are documented in the relevant papers that are cited in the next section, the suggested bibliography.

Starting from the premise that the complexity of data modeling in IR is mainly due to the complex nature of the relationship between different IR objects—documents and *auxiliary data*, meaning all those objects used to represent the semantics of the documents of the collection of interest (e.g., index terms, classification structures, thesauri)—a frame of reference becomes necessary for understanding the semantics of the relationships between IR objects. The *conceptual architecture* depicted in Fig. 1 has been proposed and used with this premise in mind.

The three levels of the structure of the conceptual architecture are (from the lowest level to the top): the document level, which contains the documents that are elementary objects of interest, the index term level, and the concept level, in which sets or classes of index terms are allocated.

A methodology has been proposed as a set of coordinated methods for indexing, querying, clustering, and browsing. The integration of these methods provides a greater effectiveness than that gained within traditional settings since the methods are designed to be interactively and directly employed by the final user.

The design process is divided into five phases, each concerned either with constructing a level of the conceptual architecture or with setting a network of links within or between levels. With the exception of the first phase, which must be performed before any other, there is no strict or unique order for performing the remaining phases.

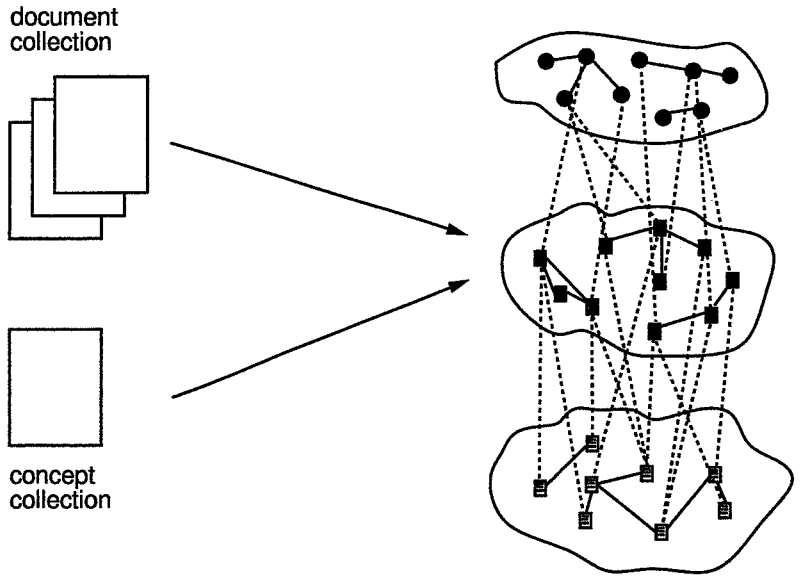

FIGURE 2. *The construction process of the network structure.*

Some phases can also be performed in parallel for a faster construction of the hypertext.

The different phases are concerned with the construction of the index terms level, the identification of the associations between documents and index terms, the construction of the concepts level, the identification of the associations between concepts, the determination of the associations between index terms and concepts, the determination of the associations between index terms, and the determination of the associations between documents.

The next step is to use the defined methods to implement the hypertext starting from the flat textual collection of documents.

The current implementation of our methods assumes that the user access the IR hypertext using a WWW browser. The bridge between our conceptual model and the *Web* is the *hypertext mark-up language (HTML)*, which enables the transformation of a flat text into a hypertext. This transformation is achieved by inserting HTML tags into the document to be transformed; that is, "authoring" the documents.

The output of the construction process is the network structure depicted in Fig. 2. Conceptually the structure is composed of nodes and links, but there are two types of nodes.

- *Hypertext nodes:* nodes representing documents, index terms, or concepts. Each hypertext node contains data and anchors to link nodes; for example, a document node includes a piece of text and two anchors, one to a list of index terms and one to a list of similar documents.

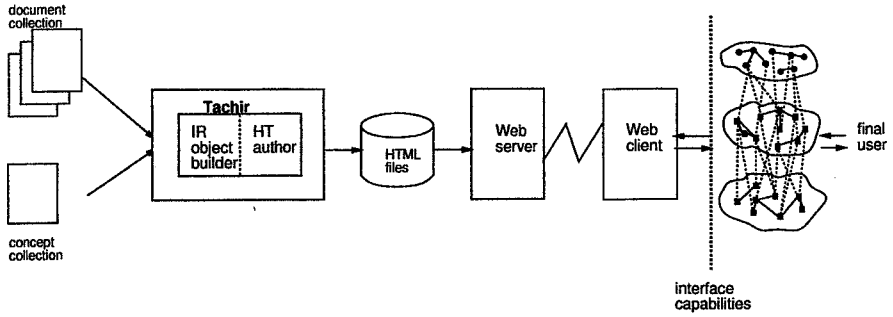

FIGURE 3. *The production of the network structure using Tachir.*

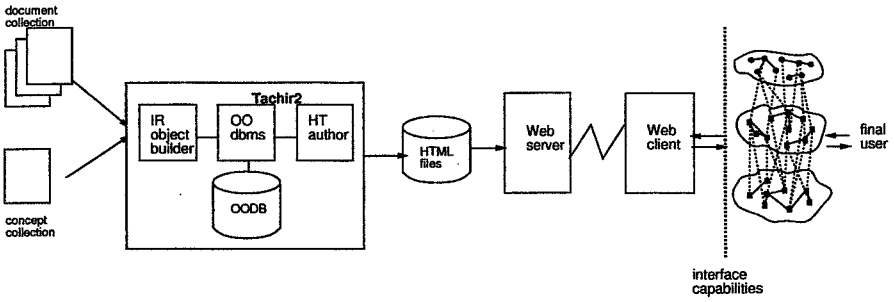

FIGURE 4. *Tachir2—a new version of Tachir supporting the update of the network structure.*

- *Link nodes:* a list of anchors to hypertext nodes; for example, a link node implementing the relationship between a document and its index terms contains a list of anchors connecting the document node to the index term nodes.

Different tools have been designed and developed over the years for the automatic construction of hypertexts. These different developments have been necessary to reach a final software architecture that is mostly independent from a specific operating environment and that is able to maintain and manage different types of document objects. The evolution of such a tool is depicted in Fig. 2, 3, 4, and 5. Figure 2 depicts the *problem*; that is, the automatic transformation of a flat collection of flat textual documents into a hypertext that is described by the conceptual schema. Figure 3 depicts the production of a network structure using TACHIR, the first tool that implements the methodology and tries to solve the problem. The hypertext is implemented as a set of HTML pages managed by the file system and accessed by a Web server from which Web clients retrieve the pages to be rendered to the final user. The main drawback of TACHIR is the difficulty of updating the hypertext whenever new material has to be inserted into the hypertext. The problem of hypertext update

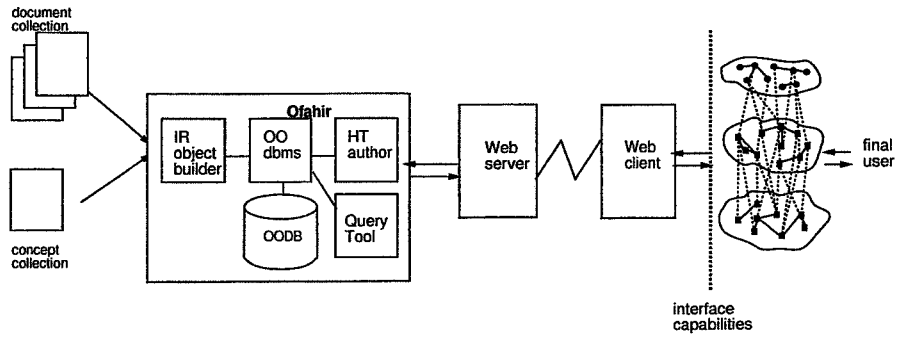

FIGURE 5. *The "on-the-fly" automatic construction of hypertexts using Ofahir.*

caused by the insertion of new documents has been addressed by the second version of TACHIR, depicted in Fig. 4. An object-oriented storage manager is employed to store indexing data, the document collection, and the set of concepts implementing the third level of the hypertext schema. The indexing process involves the new document only and the new hypertext is built starting from the database instead of the original raw data. The database cannot be queried, however, and no integration of browsing and querying is allowed. To address the problem of browsing and querying integration, TACHIR evolved to OFAHIR to allow the "on-the-fly" automatic construction of hypertexts illustrated by Fig. 5.

SUGGESTED BIBLIOGRAPHY

The IR area and the relevant problems that addresses are widely illustrated in different seeds books, among them Refs. *36–40*. The vector-space model was proposed in Salton et al. (*41*), and then established in Salton and McGill (*37*). The probabilistic model had several contributions and a long history, as shown in Refs. *42–46*. The most famous stemming algorithm is described in Porter (*47*). Document and query term weighting is addressed in many fundamental articles (*48–50, 44*).

Text clustering is discussed in general in Refs. *51, 36,* and *52*, in which the cluster hypothesis is also presented and analyzed. Text clustering has been addressed in the context of the automatic construction of IRS in Refs. *53* and *54*. Many clustering algorithms were studied in the early days of IR, and recently some efficient algorithms have been proposed to deal with the vast amount of information available in machine-readable form. The interested reader can refer to Refs. *36–38* and *40*.

Different approaches to text structuring are presented in Refs. *55, 53,* and *56*. Diverse text structure-based methods for the automatic construction of nodes and links are presented in Refs. *22–24*.

Theme extraction has been addressed by the Cornell research group, and initial

results are presented in Ref. *19*. Singhal discussed the topic in cooperation with the late Professor Salton, and more advanced results are presented in Ref. *6*.

Summarization has been addressed as early as modern indexing in the pioneering work of Luhn (*57*). In general it has been addressed by several researchers from different points of view and with different approaches. Two fairly recent works survey what has been done over the years (*58, 59*).

Passage retrieval began to be addressed some years ago (*60*). More recently, the topic has become crucial for the wide availability of digital collections of big textual documents where it becomes important to address the interested user toward a specific relevant fragment of the whole document. Some relevant works that have been published in recent years are Refs. *61–65*.

Relevant aspects concerned with hypertext structure and related measures are addressed in Refs. *25* and *20*.

Text content-based methods for the automatic construction of nodes and links have been studied by different authors and with different approaches (e.g., *21, 66, 6, 67, 18*, and *68*).

The approach for the automatic construction of hypertexts developed by the IMS group of the University of Padua has been documented (*69, 67, 70, 18*, and *4*).

Evaluation

An argument in favor of automatic hypertext construction is that it involves less subjective and partial judgments about the goodness of hypertext than a human-made construction does. It is important to note, however, that automatic tools also incorporate some subjectiveness since they are designed and implemented by human experts as well. For example, most methods for automatic hypertext construction are based on the computation of the vector cosine to assess the similarity between document nodes, but this measure is only a rough approximation of the semantic closeness between documents and is the result of the design process made by a human designer. Automatic construction can *help* in reducing the risk of inconsistency during the creation of nodes or links that may occur during a man-made hypertext construction. We stress the term help since it is still unclear whether automatic hypertext "authors" are affected by a higher or lower probability of either detecting wrong links or missing the right ones than manual authors. To say whether IR-based automatic hypertext construction is a good way to create effective hypertexts, some sort of evaluation procedures are needed.

Evaluation is a mandatory step to decide whether or not automatically constructed hypertexts and the techniques being used to build them are effective tools to support the user in retrieving information. The effectiveness of automatically constructed hypertexts strongly depends on the IR techniques used to build nodes and links. An evaluation should therefore tell us whether or not IR techniques effectively support the automatic construction of both links that represents semantic relationships and nodes that store relevant information.

The area of HIR, and specifically of automatically constructed hypertext evalua-

tion, is still in its infancy, and it is rather difficult to propose a taxonomy of different approaches and then to evaluate which methodology is better than another. In this article, we adopt a broad classification, being induced by the "observed" past research contributions, and based on two orthogonal dimensions.

1. The involvement or exclusion of "real" elements (i.e., database, collections, and users)
2. Factors to be evaluated by quantititative measures

As regards the first dimension, it is important to note that the evaluation of HIR systems is related to the issue of evaluation of IRS since an HIR system can be considered as a type of IRS. The evaluation methodologies of IRS can be split into two broad categories that correspond to the extremes of a wide range of approaches: the methodologies that take into account the user, and the ones that do not (*71*).* The same distinction can be applied to HIR evaluation. The former are usually implemented within operational environments, in which real databases managed by real systems are used by real users issuing real queries. The role of the user during evaluation is as important as the three variables (i.e., database, collections, and the user and his queries). Methodologies that ignore the role of the user during IR processes are usually implemented within a laboratory environment in which test collections managed by ad hoc prototypes are used against artificial queries, and effectiveness is measured through evaluation measures; for instance, precision and recall. The majority of the research contributions to the evaluation of automatically constructed hypertexts have been approached from the system side since

1. The user-side evaluation is more resource-consuming and can be repeated with difficulty.
2. It is quite natural to compute evaluation measures through the same software tool that generated the resulting hypertext.

With respect to the second dimension, evaluation of automatic hypertext construction methods can be approached by measuring three different but complementary factors, as illustrated in some of the research work cited and surveyed in the following:

- The *goodness of nodes in storing relevant data*, as done for example in Ref. *72*.
- The *hypertext topology*, as tested in Refs. *25* and *73–75*.
- The *degree* to which *links are able to represent the semantic relationships between nodes*. As regards this point, the following criteria to assess the quality of hypertext links have been identified (*76*):
  - *Consistency* between different hypertexts constructed by a specific tool or human experts, as presented in Refs. *74* and *73*.
  - *Soundness* of links in connecting only semantically related nodes. The criterion is analogous to the notion of precision (*77, 78, 74*).

*At the end of the previous main sections we inserted a subsection, "Suggested Bibliography," that is devoted to the most relevant contributions for automatic hypertext construction. We decided to insert such a specific bibliography because it is possible to identify different consolidated approaches for the addressed topics, and classify and assign them to the proposing "schools." In this section, we prefer to insert the relevant bibliographic citations directly into the text since evaluation is far from being a mature research area and evaluation approaches are tightly coupled to the researchers who proposed them.

- *Completeness* of links in connecting all relevant nodes as recall is the degree to which all relevant documents are retrieved (*77, 78, 74, 20*).
- *Update* of hypertext links by inserting new nodes without violating soundness and completeness (*79, 75*).

EVALUATION APPROACHES FROM THE USER SIDE

The evalution of HIR effectiveness is termed *user side* if it is carried out through operational or laboratory tests with the active and direct users' participation. Some of the proposed evaluation procedures have been designed to compare man-made hypertexts to automatically constructed ones under the assumption that the former are the "ideal" ones and so better than the latter.

Blustein et al. (*74*) report on two automatically constructed hypertext evaluation methods by measuring the *consistency*. A method aims to compare an automatic hypertext to an "ideal" one by computing correlation coefficients between the length of the paths between the documents, and the document similarity expressed through the cosine measure of the weighted vector-space model. The other method is based on the assessments given by real searchers. The latter belongs to the user-side category. The experiment is an instance of the scheme described in Tague-Sutcliffe (*71*), yet in Blustein et al. (*74*), only the methodological part has been illustrated and no results are reported. The relevance of this paper is its discussion of hypertext evaluation, particularly the following:

1. The usefulness of a hypertext can be better understood if it is compared to the correspondent plain text from which it is generated.
2. It should be clear for which population of users the hypertext is designed.
3. The significance of tests strongly depends on the degree of representativeness of the sample of users; for example, the more the population is divided, the larger the sample should be.
4. A distinction has to be made between observable variables and extraneous factors; the latter, such as interface or hypertext subject, must be controlled.
5. Because of interdocument links, the judgment about the relevance of a document depends on the judgments about the previously visited documents.

Another interesting approach is presented in Furner et al. (*73*), in which the authors report on an experimental study comparing hypertexts in which different people were asked to produce a hypertext representation of the same full-text document. The aim of the tests carried out by these authors is to measure the interlinker *consistency*; that is, the extent to which different human experts produce different hypertexts. The investigation required the calculation of measures of similarity between pairs of manually produced hypertexts. It shows that the structure of a hypertext document is crucially dependent upon the person who created the links. This important experimental result implicitly supports the study and development of methods for the automatic authoring of hypertexts, since the application of a well-founded automatic authoring method can produce a hypertext that does not incorporate just one specific subjective design and development view. Also, it is reproducible, starting from the same initial collection of flat documents. The authors conclude that they are "unable

to reject the null hypothesis that there is no positive association between inter-linker consistency and retrieval effectiveness."

Most of the research work on automatic hypertext construction is devoted to the evaluation of the quality of the links. An important methodological contribution to evaluate the *goodness of nodes* in storing relevant data is presented in Salton et al. (*72*). The authors present some methods to automatically extract summaries from text documents. The extracted summaries can in principle be used as hypertext nodes. The authors set a test procedure to compare manually extracted summaries to automatically extracted ones. The user submits a document to the system, which presents two summaries: a manual summary and an automatic one. The user judges both summaries with respect to his own notion of quality. Under the assumption that manually made summaries are the "ideal" ones, the overlap of selected automatic summaries with manual summaries is computed as a measure of the goodness of the automatic summary.

EVALUATION APPROACHES FROM THE SYSTEM SIDE

The evaluation of HIR effectiveness from the *system side* is based on the computation of quantitative measures and the simulation of retrieval processes, such as querying and browsing. These tasks are performed through the use of some test collections within a laboratory environment and without any real users' participation. This type of evaluation can consist of

- The computation of classic precision-recall measures (*77, 78*)
- The computation of measures describing the hypertext topology (*25, 73, 74*)

Although the methodologies proposed in Refs. *25 and 73* are designed for manual hypertext construction, they can be incorporated into any automatic construction algorithm to dynamically guide the process.

In Croft and Turtle (*77*) a comparison is made between a probabilistic retrieval model incorporating interdocument links to a heuristic spreading activation strategy. The aim is to improve the *completeness* and the *soundness* of the hypertext, yet the network is employed as a "low-level" device for IR rather than as a browsing tool. The main findings are that the use of hypertext links makes retrieval more effective than strategies without links. Specifically, manually constructed links such as bibliographic citations are more effective than automatically constructed ones, such as the nearest neighbor's links. The authors stress the importance of implementation since the use of hypertext links in retrieval strategies requires additional computation resources to store and process links.

Savoy evaluated the *completeness* and the *soundness* of interdocument links designed and implemented on the grounds of three types of bibliographic citations (i.e., bibliographic reference, bibliographic coupling, and co-citation), as well as links based on nearest neighbor nodes (*78*). Two test collections (i.e., Communications of the ACM (CACM) and Collection of the Institute of Scientific Information (CISI) have been employed (*80*). The former also incorporates bibliographic citations between documents. The results confirm the findings made by Croft and Turtle (*77*)

and show that links based on bibliographic citations are more effective than links based on nearest neighbors since the former are carefully inserted by the document authors.

The effectiveness of bibliographic data depends on the availability and organization of references and citations. For example, in Crestani and Melucci (*16*) the authors conducted a case study of automatic authoring by transforming a textbook into a hyper-textbook. A textbook such as that used in Crestani and Melucci (*16*) is often rich in bibliographic data, but consists of outgoing citations, and few bibliographic references occur within the same citing text paragraphs. The same holds for journal articles. The important lesson we can learn from the work presented in Savoy (*78*) and Croft and Turtle (*77*) is that if available, man-made links such as bibliographic-based links are more effective than—and are an "upper limit" to—automatically made links. It is therefore necessary to devote more research to the evaluation of automatically constructed links to understand whether or not they are effective enough whenever links made by a human expert are absent, or in what proportion the effectiveness of automatic links is less than the effectiveness of manual links.

Another approach to evaluating the quality of a hypertext is based on the computation of measures describing the *hypertext topology*. An example of such an approach is described by Botafogo et al. (*25*), in this case to evaluate the "goodness" of a hypertext that is being manually constructed by a human expert and that is based on hierarchical structures. The interest in the hierarchical hypertext structures is due to the fact that they are considered as the most effective ones to avoid the problem of disorientation from too many jumps in navigating the hypertext. The evaluation methodology is based on two types of information: (1) the indication given to the author to recover lost hierarchies and to define new ones, and (2) functions measuring the compactness and stratum that are respectively the "intrinsic connectedness of the hypertext" and "the degree the hypertext is organized so that some nodes must read before others."

The approach reported in Smeaton and Morrissey (*75*) is interesting because it is a combination of IR techniques and *hypertext topology* measures in order to control the incremental *update* of the hypertext. The approach addresses the problem of the automatic construction of a hypertext from a text using techniques from IR that are dynamically guided by an overall measure of how good the resulting hypertext is. The approach is not new from the point of view of the techniques used: the classical $tf \cdot idf$ measure of node–node similarity is used, while the goodness of the resulting hypertext is measured using the Botafogo measure of compactness. The novelty of the approach proposed is in the use of Botafogo's measure as a dynamic control structure that provides a cutoff point in the incremental addition of links in the hypertext. The resulting approach provides a technique for the selective creation of links based on a measure of the overall hypertext topology that controls the addition of a new link in relation to its influence on the overall hypertext topology.

Furthermore, in Smeaton (*20*), the use of the *compactness* measure proposed by Botafogo for the construction of hypertexts is analyzed. The analysis is performed by evaluating the compactness measure using a Web robot on four different hypertexts. These hypertexts are based on four quite different topologies and are either manually (two of them) or automatically constructed (the other two), with links reflecting only the structure of the document (first-order hypertexts) and/or links reflecting content

similarity between nodes (second-order hypertexts). The effect on the compactness measure of the different topologies is analyzed and some interesting conclusions are drawn. The major conclusion, however, is that the compactness measure is not particularly useful in guiding the automatic construction of a large hypertext, but it could be useful for the automatic authoring of subparts of the hypertext.

The major contribution of these works is in proposing guidelines to control the automatic construction of the hypertext. What remains to be discussed is the relevance of the hypertext topology to hypertext effectiveness. As the authors highlighted, some very compact hypertexts may result in the user being disoriented because of too many links. Moreover, a compact hypertext is not always desirable; in a hypertext with a large number of links the user can be helped browse by the provision of more information about links, such as, for example, information about the link types.

The incorporation of *hypertext topology*-based measures has been also proposed by Furner et al. (*73*). Their aims are different from those reported in Botafogo et al. (*25*), however. They do in fact employ some graph theory results, such as adjacency, distance, and converted distance to compare the hypertexts produced by different authors.

In Blustein et al. (*74*) an evaluation method based on the computation of numerical measures is presented. The aim was to study the relationships between *soundness* and *completeness* of the hypertext links. In particular the authors studied the correlation between the similarity function used to decide whether or not to insert a link between two documents as well as the length of the paths between documents. There are two main conclusions: (1) similarity value and path length are inversely correlated (i.e., highly similar documents are connected by short paths), and (2) the correlation becomes lower as the number of outgoing links increases. This means that considering document nodes with a single out-degree leads to sound but incomplete hypertexts. On the contrary, nodes with several outgoing links form complete but imprecise hypertexts.

OPEN PROBLEMS AND FUTURE DIRECTIONS

The high degree of complexity of automatic hypertext construction is due to the uncertainty added by the algorithms that implement the techniques employed to detect links and nodes. Such uncertainty is analogous to the one affecting IRS that try to retrieve all and only the relevant documents, thus minimizing the probability of retrieving nonrelevant documents. Some questions about the evaluation of automatic hypertext construction techniques seem to remain partially unsolved.

1. The resulting hypertext should be tested to see if it is useful for IR purposes, both from a system and user's point of view. About the former, the hypertext could be used as a low-level data structure to automatically enhance the query-based retrieval algorithm; for example, by adding to the retrieved documents the similar ones. Although good from an IR point of view such hypertext might not be good for user browsing since the same interdocument links might have a cryptic semantics if displayed on a screen.
2. Automatically constructed hypertexts can be employed as highly interactive IRS, and therefore one should take into account the role of the user whenever an evaluation is designed. This is because the retrieval process is done by both the system and the final

user, who is asked to make appropriate browsing selections to retrieve relevant documents. Retrieval effectiveness therefore depends on how well the user browses the hypertext, and in the last analysis how well the automatic hypertext construction process performs.

3. Are IR techniques effective for documents covering heterogeneous subjects? The problem requires further attention, but it seems possible to anticipate that the effectiveness of a technique for the automatic construction of hypertexts depends on the spread of the subjects of the documents.
4. The goodness of the resulting hypertext is related to the fine-tuning of some parameters, such as the similarity threshold. Is there an automatic technique for setting these parameters? How do they affect the effectiveness of the hypertext?
5. Laboratory evaluation of automatically constructed hypertext is more difficult since we lack the test collections that are necessary to carry out laboratory and computer-based tests. Test collections for evaluating automatic hypertext construction methods should include, among the other things, relevance assessments about links other than about documents to evaluate if a method is able to detect all links and not only the significant links between each pair of documents.

The use of classical IR evaluation methodologies within the evaluation of HIR systems is still a topic of discussion. Indeed, an HIR environment is more complex than an IR environment for many reasons.

1. First of all, HIRs are *networks* of documents and auxiliary data rather than flat collections of flat data. Evaluation must take into account the fact that the hypertext links represent direct or indirect semantic relationships between documents, and therefore the relevance of a document with respect to a query is dependent on both directly and indirectly linked documents.
2. Evaluation measures used in IR in a laboratory setting (e.g., precision and recall) are only partially usable in the evaluation of characteristics of hypertext IRSs since these measures are based on the assumption of unlinked documents and auxiliary data. New measures should be defined to take into account the presence of a network of nodes.
3. Evaluation methodologies for IRSs are designed for query-based search processes. On the contrary, HIRs are employed to browse and query the document base in an integrated way. One should in general effectively evaluate the new HIR system capabilities, and in particular address the evaluation of those functions (i.e., integrated querying and browsing) that have been automatically developed and made available to the users.

REFERENCES

1. V. Bush, "As We May Think." *Atlantic Monthly,* **176**(1), 101–108 (1945).
2. T. Berners-Lee, R. Cailliau, A. Luotonen, H. Nielsen, and A. Secret, "The World-Wide Web." *Commun. ACM,* **37**(8), 76–82 (1994).
3. A. Mendelzon, G. Mihaila, and T. Milo, "Quering the World Wide Web." *Internat. J. Dig. Libr.,* **1**, 54–67 (1997).
4. M. Agosti, L. Benfante, and M. Melucci, "OFAHIR: On-the-Fly Automatic Authoring of Hypertexts for Information Retrieval," *Proceedings of the 7th IFIP,* 1997, EPFL Press, Leysin, Switzerland, pp. 129–154.
5. M. Hearst and C. Karadi, "Cat-a-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results Using a Large Category Hierarchy," *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR),* Philadelphia, 1997, pp. 246–256.
6. G. Salton, J. Allan, C. Buckley, and A. Singhal, "Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts," in *Information Retrieval and Hypertext,* M. Agosti and A. Smeaton, eds. Kluwer Academic Publishers, Boston, MA, 1996, pp. 51–73.

7. M. Agosti, "Is Hypertext a New Model of Information Retrieval?" *Proceedings of the 12th International Online Information Meeting,* vol. 1, Learned Information, Oxford, U.K., 1988, pp. 57–62.
8. M. Agosti, "New Potentiality of Hypertext Systems in Information Retrieval Operations," in *Human Aspects in Computing,* H. Bullinger, ed. Elsevier Science, Amsterdam, 1991, pp. 317–321.
9. W. Croft and R. Thompson, "I^3R: A New Approach to the Design of Document Retrieval Systems." *JASIS,* **38**(6), 389–404 (1987).
10. M. Agosti, ed., Special issue on hypertext and information retrieval. *Info. Proc. Mgt.,* **29**(3), (1993).
11. M. Agosti and A. Smeaton, eds. *Information Retrieval and Hypertext,* Kluwer Academic Publishers, Boston, 1996.
12. J. Smith and D. Smith, "Database Abstractions: Aggregation." *Commun. ACM,* **20**(6), 405–413 (1977).
13. U. Schiel, "Abstraction in Semantic Networks: Axiom Schemata for Generalization, Aggregation and Grouping." *SIGART Newsl.,* **107**, 25–26 (1989).
14. I. Aalbersberg, "Incremental Relevance Feedback," *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR),* Copenhagen, 1992, pp. 11–21.
15. N. Belkin, P. Marchetti, and C. Cool, "BRAQUE: Design of an Interface to Support User Interaction in Information Retrieval." *Info. Proc. Mgt.,* **29**(3), 325–344 (1993).
16. F. Crestani and M. Melucci, "A Case Study of Automatic Authoring: From a Textbook to a Hyper-Textbook." *Data Knowl. Eng.,* **27**(1), 1–30 (1998).
17. M. Bieber and T. Isakowitz, "Designing Hypermedia Applications." *Commun. ACM,* **38**(8), 26–33 (1995).
18. M. Agosti, F. Crestani, and M. Melucci, "Design and Implementation of a Tool for the Automatic Construction of Hypertexts for Information Retrieval." *Info. Proc. Mgt.,* **32**(4), 459–476 (1996).
19. G. Salton, J. Allan, C. Buckley, and A. Singhal, "Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts." *Science,* 264 (1994).
20. A. Smeaton, "Building Hypertext Under the Influence of Topology Metrics." *Proceedings of IWHD Conference,* Montpellier, France, 1995.
21. J. Allan, "Building Hypertexts Using Information Retrieval." *Info. Proc. Mgt.,* **33**(2), 145–159 (1997).
22. R. Rada, W. Wang, and A. Birchall, "Retrieval Hierarchies in Hypertext." *Info. Proc. Mgt.,* **29**(3), 359–371 (1993).
23. R. Furuta, C. Plaisant, and B. Schneiderman, "Automatically Transforming Regularly Structured Linear Documents into Hypertext." *Electronic Pub.,* **4**(2), 211–229 (1989).
24. J. Coombs, "Hypertext, Full-Text, and Automatic Linking," *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR),* Brussels, 1990, pp. 83–98.
25. R. Botafogo, E. Rivlin, and B. Shneiderman, "Structural Analysis of Hypertext: Identifying Hierarchies and Useful Metrics." *ACM Transac. Info. Syst.,* **10**(2), 142–180 (1992).
26. F. Garzotto, P. Paolini, and D. Schwabe, "HDM: A Model-based Approach to Hypertext Application Design." *ACM Transac. Info. Syst.,* **11**(1), 1–26 (1993).
27. F. Halasz and M. Schwartz, "The Dexter Hypertext." *Commun. ACM,* **37**(2), 30–39 (1994).
28. D. Lucarella and A. Zanzi, "Information Modelling and Retrieval in Hypermedia Systems," in *Information Retrieval and Hypertext,* Chap. 6, M. Agosti and A. Smeaton, eds. Kluwer Academic Publishers, Boston, MA, 1996, pp. 121–138.
29. Y. Chiaramella and A. Kheirbek, "An Integrated Model for Hypermedia and Information Retrieval," in *Information Retrieval and Hypertext,* Chap. 7, M. Agosti and A. Smeaton, eds. Kluwer Academic Publishers, Boston, MA, 1996, pp. 139–178.
30. D. Hull and G. Grefenstette, "Querying Across Languages: A Dictionary-based Approach to Multilingual Information Retrieval," in *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR),* H. Frei, D. Harman, P. Schäuble, and R. Wilkinson, eds. ACM Press, New York, 1996, pp. 49–57.
31. W. Kraaij and R. Pohlmann, "Evaluation of a Dutch Stemming Algorithm," in *The New Review of Document and Text Management,* vol. 1, J. Rowley, ed. Taylor Graham, London, 1995, pp. 25–43.
32. P. Sheridan and J. Ballerini, "Experiments in Multilingual Information Retrieval Using the SPIDER System," *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR),* Zurich, Switzerland, 1996, pp. 58–65.
33. P. Sheridan, M. Wechsler, and P. Schäuble, "Cross-Language Speech Retrieval," *Proceedings of the*

ACM International Conference on Research and Development in Information Retrieval (SIGIR), Philadelphia, 1997, pp. 99–108.

34. G. Salton, "Automatic Processing of Foreign Language Documents." *JASIS,* **21**, 187–194 (1970).
35. M. Davis and T. Dunning, "A TREC Evaluation of Query Translation Methods for Multi-Lingual Text Retrieval," *Proceedings of Text Retrieval Conference (TREC)*, Gaithersburg, MD, 1996.
36. C. van Rijsbergen, *Information Retrieval,* 2nd ed., Butterworths, London, 1979.
37. G. Salton and M. McGill, *Introduction to Modern Information Retrieval,* McGraw-Hill, New York, 1983.
38. W. Frakes and R. Baeza-Yates, eds., *Information Retrieval: Data Structures and Algorithms,* Prentice Hall, Englewood Cliffs, NJ, 1992.
39. F. Lancaster and A. Warner, *Information Retrieval Today,* Information Resources Press, Arlington, VA, 1993.
40. K. Sparck Jones and P. Willett, *Readings in Information Retrieval,* Morgan Kaufmann, San Francisco, CA, 1997.
41. G. Salton, C. Yang, and C. Yu, "A Theory of Term Importance in Automatic Text Analysis." *JASIS,* **26**(1), 33–44 (1975).
42. M. Maron and J. Kuhns, "On Relevance, Probabilistic Indexing and Retrieval." *J. ACM,* **7**, 216–244 (1960).
43. S. Robertson, "The Probability Ranking Principle in Information Retrieval." *J. of Doc.,* **33**(4), 294–304 (1977).
44. S. Robertson and K. Sparck Jones, "Relevance Weighting of Search Terms." *JASIS,* **27**, 129–146 (1976).
45. S. Robertson, C. van Rijsbergen, and M. Porter, *Probabilistic Models of Indexing and Searching,* Chap. 4, Butterworths, 1981, pp. 35–55.
46. F. Crestani, I. Campbell, M. Lalmas, and C. van Rijsbergen, "Is This Document Relevant?... Probably," in *A Survey of Probabilistic Models in Information Retrieval,* Department of Computing Science, University of Georgetown, U.K. 1997.
47. M. Porter, "An Algorithm for Suffix Stripping." *Program,* **14**(3), 130–137 (1980).
48. G. Salton and C. Buckley, "Term Weighting Approaches in Automatic Text Retrieval." *Info. Proc. Mgt.,* **24**(5), 513–523 (1987).
49. K. Sparck Jones, "Experiments in Relevance Weighting of Search Terms." *Info. Proc. Mgt.,* **15**(3), 133–144 (1979).
50. W. Croft and D. Harper, "Using Probabilistic Models of Document Retrieval Without Relevance Information." *J. Doc.,* **35**, 285–295 (1979).
51. C. van Rijsbergen and K. Sparck-Jones, "A Test for the Separation of Relevant and Non-Relevant Documents in Experimental Retrieval Collections." *J. Doc.,* **29**, 251–257 (1973).
52. A. Griffitths, H. Luckhurts, and P. Willett, "Using Inter-Document Similarity Information in Document Retrieval Systems." *JASIS,* **37**, 3–11 (1986).
53. G. Salton and C. Buckley, "Automatic Text Structuring and Retrieval—Experiments in Automatic Encyclopedia Searching," *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR),* Chicago, 1991, pp. 21–30.
54. G. Salton, J. Allan, and A. Singhal, "Automatic Text Decomposition and Structuring." *Info. Proc. Mgt.,* **32**(2), 127–139 (1996).
55. M. Hearst and C. Plaunt, "Subtopic Structuring for Full-Length Document Access," *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, Pittsburgh, PA, R. Korfhage, E. Rasmussen, and P. Willett, eds. 1993, pp. 59–68.
56. G. Salton, J. Allan, and C. Buckley, "Automatic Text Structuring of Text Files." *Electronic Pub.,* **5**(1), 1–17 (1992).
57. H. Luhn, "The Automatic Creation of Literature Abstracts." *IBM J. R. Dev.,* **2**(2), 159–165 (1958).
58. K. Sparck Jones and B. Endres-Niggemeyer, eds., Special issue on summarizing text. **31**(5), (1995).
59. C. Paice, "Automatic Abstracting," in *Encyclopedia of Library and Information Science,* vol. 53, Dekker, New York, pp. 16–27.
60. J. O'Connor, "Answer Passage Retrieval by Text Searching." *JASIS,* **32**(4), 227–239 (1980).
61. G. Salton, J. Allan, and C. Buckley, "Approaches to Passage Retrieval in Full Text Information

Systems," *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR),* R. Korfhage, E. Rasmussen, and P. Willett, eds., Pittsburgh, 1993, pp. 49–58.

62. J. Callan, "Passage-Level Evidence in Document Retrieval," *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR),* Dublin, 1994, pp. 302–310.
63. E. Mittendorf and P. Schäuble, "Document and Passage Retrieval Based on Hidden Markov Model," *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR),* Dublin, 1994, pp. 318–327.
64. R. Wilkinson, "Effective Retrieval of Structured Documents," *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR),* Dublin, 1994, pp. 311–317.
65. M. Melucci, "Passage Retrieval: A Probabilistic Technique." *Info. Proc. Mgt.,* **34**(1), 43–67 (1998).
66. M. Agosti and J. Allan, eds., Special issue on methods and tools for the automatic construction of hypertexts. *Info. Proc. Mgt.,* **33**(2), (1997).
67. M. Agosti and F. Crestani, "A Methodology for the Automatic Construction of a Hypertext for Information Retrieval," *Proceedings of the ACM Symposium on Applied Computing,* Indianapolis, 1993, pp. 745–753.
68. M. Melucci, *Costruzione automatica di ipertesti,* Ph.D. thesis, University of Padova, Department of Electronics and Computer Science, Padova, Italy, 1996, (in Italian).
69. M. Agosti, F. Crestani, G. Gradenigo, and P. Mattiello, An Approach for the Conceptual Modelling of IR Auxiliary Data," *Proceedings of the 9th Annual IEEE International Phoenix Conference on Computers and Communications,* Scottsdale, AZ, 1990, pp. 500–505.
70. M. Agosti, F. Crestani, and M. Melucci, "Automatic Authoring and Construction of Hypermedia for Information Retrieval." *ACM Multimedia Syst.,* **3**, 15–24 (1995).
71. J. Tague-Sutcliffe, "The pragmatics of Information Retrieval Experimentation, Revisited." *Info. Proc. Mgt.,* **28**(4), 467–490 (1992).
72. G. Salton, A. Singhal, M. Mitra, and C. Buckley, "Automatic Text Structuring and Summarization." *Info. Proc. Mgt.*, **33**(2), 193–207 (1997).
73. J. Furner, D. Ellis, and P. Willett, "The Representation and Comparison of Hypertext Structures Using Graphs, in M. Agosti and A. Smeaton, eds. *Info. Retrieval and Hypertext,* Kluwer Academic Publishers, Boston, MA, 1996, pp. 75–96.
74. J. Blustein, R. Webber, and J. Tague-Sutcliffe, "Methods for Evaluating the Quality of Hypertext Links." *Info. Proc. Mgt.,* **33**(2), 255–271 (1997).
75. A. Smeaton and P. Morrissey, (1995). "Experiments on the Automatic Construction of Hypertext from Text," technical report, working paper CA-0295, Dublin City University, School of Computer Applications, Dublin, Ireland.
76. P. Thistlewaite, "Automatic Constructions and Management of Large Open Webs." *Info. Proc. Mgt.,* **33**(2), 161–173 (1997).
77. W. Croft and H. Turtle, "Retrieval Strategies for Hypertext." *Info. Proc. Mgt.,* **29**(3), 313–324 (1993).
78. J. Savoy, "Ranking Schemes in Hybrid Boolean Systems: A New Approach. *JASIS,* **48**(3), 235–253 (1997).
79. M. Agosti, L. Benfante, and M. Melucci, "OFAHIR: On-the-Fly Automatic Authoring of Hypertexts for Information Retrieval," in *Data Mining and Reverse Engineering: Searching for Semantics,* S. Spaccapietra and F. Maryanski, eds., IFIP, Chapman and Hall, London, U.K., 1998, pp. 269–300.
80. E. Fox, "Characterization of Two New Experimental Collections in Computer and Information Science Containing Textual and Bibliographic Concepts," technical report TR83-561, Computer Science Department, Cornell University, Ithaca, NY, 1983.

MARISTELLA AGOSTI
MASSIMO MELUCCI