OFAHIR: “on-the-fly” automatic authoring of hypertexts for information retrieval

M. Agosti, L. Benfante, and M. Melucci
Università di Padova
Dipartimento di Elettronica e Informatica
Via Gradenigo n. 6/a, 35131 Padova Italy
{agosti,benfante,melo}@dei.unipd.it

Abstract

This paper presents the OFAHIR methodology that aims at providing the final user an integrated environment for gathering useful information from a collection of managed digital documents. The methodology is a set of co-ordinated methods for indexing, querying, clustering, and browsing; this set constitutes an integrated **associative retrieval system model**. This is the model for a system that enables the user to access a collection of information objects by integrating the processes of querying and browsing; the managed objects are linked making use of semantic associations that are built through indexing and clustering mechanisms. The OFAHIR methodology is the first step towards the definition of a general and comprehensive associative retrieval model for information gathering. Together with the methodology also a system tool that implements it is presented in the paper. This system can work in a World Wide Web environment since the components performing the data retrieval have been implemented according to standard protocols. Through any available Web client a user can pose queries to and/or browse the managed document digital collection. Performance and evaluation aspects are also addressed in the paper, together with aspects that are going to be addressed in the future.

Keywords

Automatic Indexing, Classification, Clustering, Advanced Query Models, Tools for Automatic Construction of Hypertexts, Automatic Extraction of Semantic Information from Documents.

1 INTRODUCTION

The growing availability of **unstructured** and **full-text** data in machine readable form is being observed in recent years. The difficulty in accessing

this kind of collections is due to the absence of a schema both representing the semantics and supporting the user during the retrieval process. The World Wide Web phenomenon, and the advent of collections stored on CD-ROM have given rise to the problem of managing and accessing collections of full-text articles, technical reports, and book-size documents which are made available to the user through browsing or searching tools.

Classical models for information management and retrieval seem to be inadequate to enable researchers to face this new situation because traditional information retrieval (IR) models are no longer suitable to manage the intrinsic uncertainty in accessing these collections.

Hypertexts for IR (HIR) have been recently proposed as alternative models for the management and retrieval of information (Agosti and Smeaton 1996). What characterises HIR models is the integration of traditional retrieval capabilities, such as field-based or full-text searching, with hypertext capabilities, such as browsing. However the database underlying a hypertext for IR needs to be constructed to make a HIR effective as a tool for managing and accessing information. The complexity of **hypertext construction** has been the reason why the issue of **automatic authoring** has been addressed by researchers since the early days of hypertexts. The increasing availability of on-line text document collections, whose size is too large to enable a manual authoring and construction of the hypertext, has been the stimulus for which fully automatic, or partially automatic techniques are currently being studied and implemented (Agosti and Allan 1997). It is necessary to note that at present just few tools are available that support a wide range of facilities as would be necessary for the full automatic construction of hypertexts for IR (see for example the survey in (Agosti, Crestani and Melucci 1997)).

The research work we carried out in the area of automatic authoring (Agosti, Crestani and Melucci 1996) has confirmed that techniques for automatic authoring of hypertexts are necessary solutions to the problem of understanding and discovering the semantics of data stored in unstructured and large collections of digital information. Our previous results are a starting point for the definition of advanced models which represent a breakthrough in information management and retrieval research. This type of advanced models is needed to integrate different paradigms of information access, such as querying and browsing.

An associative retrieval model can be a suitable solution to integrate querying and browsing paradigms since associative models enable to capture the rich semantics of a large and unstructured full-text collection thanks to the intensive use of networks expressing such a semantics. However no definitive answers have been given by now both in the definition of such models, and in the design of systems implementing them.

To address the issue of developing an associative retrieval model is important in the context of a digital library architecture and project, as in the context of the **DL-IMS Initiative** that is under development at the Depart-

ment of Electronics and Informatics of the University of Padua. In fact, an associative retrieval model can be used as the underlying and reference model for the organisation of general digital information collections as it is necessary to manage in the context of a digital library implementation. Where by **Digital Library (DL)** is intended an organised collection of digital information, which combines the structuring and gathering of information, which libraries and archives have always done, with the digital representation that computers have made possible (as defined in (Lesk 1997)).

This paper presents the OFAHIR (“On-the-Fly automatic Authoring of Hypertexts for IR”) methodology, which constitutes the component for the automatic and real-time construction of hypertext structures for a digital library implementation. The paper is organised as follows: Section 2 illustrates what is meant in general by associative retrieval. Section 3 shortly illustrates all issues that are related to a methodology for the construction of hypertexts for information gathering, and to a system implementation that makes use of such a methodology. Section 4 describes the OFAHIR methodology itself. In particular, is addressed the use that is made in OFAHIR of clustering for discovering and using the semantics hidden in documents indexes. Section 5 describes the general structure of a hypertext that is automatically built making use of the OFAHIR methodology. Section 6 gives an account of the tool that has been developed to implement the OFAHIR methodology. An initial account on the performance of the tool and evaluation of possible automatic implementations of hypertexts are addressed in Section 7. Conclusions we draw from the work presented in the paper and future directions that we intend to explore are presented in Section 8.

2 ASSOCIATIVE RETRIEVAL

Associative retrieval is a form of information management that can be used by a user in the information retrieval process. The idea behind this form of information management is that of making possible to retrieve relevant information by retrieving information objects that are “associated” with some information the user knows about, or he has already retrieved and that he knows to be relevant, or that are pointed out to the user as a result of a query. The concept of associative retrieval dates back to the Salton’s early work in IR, when he proposed a method for associating documents using bibliographic citations (Salton 1963), and algorithms for computing inter-term statistical similarities to estimate the semantic associations between thesaurus items (Salton 1968).

What distinguishes associative retrieval models from classical IR models is the use of networks connecting **different** types of information objects stored in large digital collections. Those networks are important in supporting the final user in the process of retrieving information objects from large collections of digital documents.

Many are the types of objects that can be associated through networks to be accessed by the user seeking for information which are relevant to his informative need. It is possible to use a very general distinction and to distinguish different information media, although up until now the wider experience of management has been reached in the management of full text documents, where by text in most cases it is meant English text only. Within the domain of textual objects we have:

- a complete textual document;
- a single specific part of a document to be extracted through passage retrieval methods;
- relevant terms extracted from the documents of a collection;
- index terms, dictionaries and/or lists of index terms;
- concepts, and information structures used to manage and relate concepts such as a thesaurus or a classification system.

Associations that express informative relations useful to the user looking for information can exist among different types of information objects. But the associations between information objects can either be **static** and already built and existing when the user is gathering for information (e.g. query session time) or **dynamic** and determined at run time.

Static links: Static links are built using associations among information objects and are created before the gathering for information user's session. The process of static links construction makes use of semantic relationships between information objects, such as for example thesaurus-like relationships among index terms, bibliographic citations among documents, or statistical similarity associations among documents or terms.

Dynamic links: In the case of dynamic links, on the contrary, the system determines associations between information objects interacting with the user during the information gathering process. As an example, the system answers a user's query building a cluster of documents that are calculated to be relevant to the query.

3 RELATED ISSUES

The idea of inter-connecting information objects has been extensively studied in the IR area. We have already cited Salton's work in Section 2, clustering and relevance feedback mechanisms are two other most notable examples. However, few experimental IR systems provide clustering or relevance feedback since these methods are still at research level, and evidence of their general effectiveness is ambiguous.

Figure 1 depicts the "U"-shaped model that a classical IR system imple-

ments. Indexing, matching, and feedback are the processes of an IR system, and they are depicted as ovals in the figure. Arrows between ovals depict the input-output data flows between processes. The main problem of a traditional IR system is that its interaction with the user is very poor since querying is the only way the user can access the stored information, and he cannot directly use any sort of links between the managed information objects. In particular, the index describing the semantic content of both documents and queries is hidden to the user.

Figure 1 The model of an Information Retrieval system

Some advanced systems implement **relevance feedback** to automatically expand or reduce the result of a user's query asking the user to mark retrieved documents as relevant or non-relevant, in this way enabling the system to use the information given by the user on the relevance of the documents previously retrieved to retrieve a new set of documents that are calculated as more relevant to the initial query than the previously retrieved ones (Maron and Kuhns 1960, Rocchio 1971, Harman 1992). The relevance feedback underlying assumption is that query terms are more concentrated within relevant documents, and less concentrated within non-relevant documents, as non-query terms are supposed to be more present in non-relevant documents. It has been shown that relevance feedback can be very effective (Sparck Jones 1979, Salton and Buckley 1990), but there are also some limitations. One of these limitations is that the effectiveness of relevance feedback depends on the degree of document and/or query exhaustivity, i.e. the extent to which query and/or

document terms describe the topics expressed through the query as a very short for of expressing user's information needs. For example, if query terms are too general or the query is too short, it is difficult to distinguish relevant documents from non-relevant documents (Kwok 1996), because short queries lack of variety of terms and it is impossible to estimate the importance of a query term using its intra-query frequency. This situation, often called "short query" situation occurs very frequently within information seeking environments, such as the search engines running in the Web which, on the other hand, do rarely implement relevance feedback.

Clustering in IR means both the mechanism of grouping information objects in classes according to a given criterion, and the result of the application of such a mechanism. In general, the criterion is that of **similarity** among cluster elements. Clustering is one of the most used techniques to implement classification for IR purposes, because it enables the discovering of semantic relationships associating objects and provides the user with a representation of the document collection semantics; furthermore it allows the system to effectively use the relationships between objects during the processes of indexing and querying.

Clustering is applied to use and make explicit the semantic relationships occurring between documents or index terms; when it is used on documents, it is called "document clustering", when it is used on index terms, it is referred as "term clustering". **Document clustering** aims to group together documents that have something in common; for example, a document cluster can be built identifying the group of documents that have in common a set of index terms and the cardinality of the set is over a stated threshold. The effectiveness of document clustering depends on the degree to which the **cluster hypothesis** is satisfied. The cluster hypothesis states that the documents relevant to a given query are grouped within the same cluster (van Rijsbergen 1979). Thus, an IR system using document clustering matches a query against a document cluster rather than an individual document: the more the cluster includes relevant documents, the smaller the proportion of non-relevant documents that are retrieved. Conversely, **term clustering** is used to group together index terms on the basis of a specific criterion; for example, a cluster of terms can be built by the terms that occurs on the same documents and the number of these documents where they occur is over a stated/given threshold value. The effectiveness of term clustering depends on the degree to which the **association hypothesis** is satisfied. The association hypothesis states that "if an index term is good at discriminating relevant from non-relevant documents, then any closely associated index term is also likely to be good at this" (van Rijsbergen 1979). Term clustering is employed in query modification as a mean to perform a relevance feedback mechanism to expand a query through the insertion of the cluster of index terms that are similar to a query term. This allows for incorporating in the query those terms occurring within the relevant documents, but not included in the query by the

user. In principle, clustering can be employed in an effective way within an interactive environment; a relevant example is the use suggested and showed by Harman in (Harman 1988), where a perfect selection is simulated and carried out by the user of index terms to be added to the query.

Experiments in both document clustering (Willett 1988, Griffiths, Luckhursts and Willett 1986) and term clustering (Sparck Jones 1971) have shown that the effectiveness of these methods mostly depends on: characteristics of the used document collection, size of documents, and size of clusters. In particular, it has been shown that clustering can be a very effective method to retrieve information if it is implemented in a system that provides an appropriate user interface (Hearst and Karadi 1997). Furthermore, the increase of performance of current computers can make feasible clustering techniques that were difficult to efficiently implement some years ago, a fact that made the use of these advanced methods difficult in past commercial systems.

Figure 2 The use of adaptation within the automatic construction of hypertexts

The process of user-system interaction is therefore more complex than the one depicted in Figure 1, and a more sophisticated system model has to be taken into account. Few experimental IR systems provide with an integrated environment where both document and index term clustering are performed together to give the user a more semantically reach representation than that given within a classical IR setting. A new class of system models has to be designed to give the user a more effective interaction with the system than that provided by classical IR systems; the interaction can be enhanced in

different ways, one way of implementing it is by taking advantage of the data automatically managed and stored by an IR system, and making these data available and using them to build associations between different types of information objects.

Information retrieval is intrinsically and interactive process, since any IR system brings together relevant information for a user through a number of interactive processes. **User modelling** is therefore important since it places the user at the centre of the retrieval process, and it is orthogonal to the classical system-centred approach. This issue is stressed, for example, in (Saracevic, Spink and Wu 1997) where the **system-side** and **user-side** approaches to IR are outlined among the others. **System-side** approaches are historically very important since the most studied methods, such as relevance feedback and query modification date back to sixties. The area of **user-side** approaches is very wide since user is much more complex than a computing tool such as an IR system. To name but a few, user-side approaches are presented in (Ingwersen 1992, Belkin, Oddy and Brooks 1992). Our work needs to be placed on the system-side, since our main aim is the system-side design of an IR environment, which is based on the automatic construction of hypertexts. The issues of user modelling are outside the focus of this paper.

Together with user modelling, **adaptive hypertext and hypermedia** are relevant to this work since they deal with nodes and links adaptation to a given user model (Brusilovsky 1996). The role of adaptation methods within IR hypertexts is helping the user choose of “good” links. The methodology presented in this paper can in principle be integrated to an adaptive mechanism since it can also automatically and dynamically adapt link construction and node presentation to a user model that can be modified at search-time, as depicted in Figure 2.

4 OFAHIR METHODOLOGY

The OFAHIR methodology is a set of co-ordinated methods: indexing, querying, clustering, and browsing which constitute an integrated **associative retrieval system model**. This is the model of a system that enables the user to access a collection of information objects by integrating the processes of querying and browsing the objects linked by semantic associations exploited by indexing and clustering. The OFAHIR methodology aims to provide the user with these methods within a coherent and integrated framework. The integration of these methods would provide a greater effectiveness than that gained within traditional setting since the methods are designed to be interactively and directly employed by the final user. The OFAHIR methodology is the first step towards the definition of an associative retrieval model which enables the user to access a collection of information objects by integrating querying and browsing.

Figure 3 depicts the OFAHIR system model. The OFAHIR system inte-

Figure 3 The model of the OFAHIR system

grates, rather than substitutes the IR system by inserting two additional processes, i.e. clustering and browsing, and linking them to the IR processes both through the index produced as output by the indexing process, and the set of retrieved documents produced as result of the process of query-documents matching. The **key idea** is that (1) the data stored within the index are the basis to automatically build a two-level hypertext where the first level is the one of documents, and the second level is the one of index terms describing the documents, and (2) the retrieval result is an entry-point to start browsing.

In the following we give a description of the OFAHIR methods: indexing, querying, clustering, and browsing, which are performed in a partial order as depicted in Figure 4.

Indexing must be performed first since it provides with the basic data to perform all the other tasks. Indexing has to be performed several times whenever a new set of documents has to be inserted into the collection. After indexing has been performed, the user can either submit a query to retrieve documents that are relevant to his informative needs, or start browsing. If a query is submitted, the cluster of relevant documents is dynamically built, and is presented to the user. Each document is ranked by a similarity value representing a measure of association between the query and the document. Retrieved documents are information objects satisfying the user's informative need, but if the user is not fully satisfied, they can also work as entry points for browsing other linked information objects, i.e. similar documents or index terms describing the semantic content. Then clustering is performed to dynamically build either the cluster of similar documents, or the cluster of index

Figure 4 How OFAHIR tasks are related

terms. The user can perform again querying, or continue browsing to access clusters of terms or of documents, as showed in Figure 5.

4.1 Indexing

Indexing consists of the association with each document of a set of descriptors representing the semantic content of the document itself. It is the core process of an IR system since search efficiency and effectiveness depend most on it. It is by means of this process that individual or groups of terms found in documents become index terms, assuming a representational power that places them on a higher level of abstraction than the documents. Indexing can be either an automatic or a manual task according to the size of the document collection to be indexed, to the required degree of exhaustivity or specificity, and to the availability of computational resources. The reader can refer either to (Salton and McGill 1983) (Chapter 3) or (van Rijsbergen 1979) (Chapter 2) for an exhaustive presentation of the topic.

The implementation model underlying OFAHIR is the weighted vector-space model (Salton and McGill 1983). According to the vector-space model, an information object x is described as a vector $\vec{x} = (x_1, \dots, x_T)$, where x_i is an object descriptor belonging to a set of T descriptors. Therefore, document n is described as $\vec{n} = (n_1, \dots, n_K)$, where K is the total number of index terms, and $n_k = 1$ if term k occurs in n , 0 otherwise. Analogously, term k is described as $\vec{k} = (k_1, \dots, k_N)$, where $k_n = 1$ if document n includes term k , 0 otherwise.

Figure 5 OFAHIR system-user interaction

For each document-term pair (n, k) , one can compute the $tf \times idf$ -based weight $w(n, k)$ (Sparck Jones 1972). The $tf \times idf$ weighting scheme can be expressed as $w(n, k) = tf_{kn} \log \frac{N}{df_k}$ where tf_{kn} is the number of occurrences of term k within document n , df_k is the number of documents where term k occurs, and N is the total number of documents. As w can take null values whenever the logarithm is null, even if tf is positive, a slight different formula can be employed to compute the $tf \times idf$ weight, that is $w(n, k) = tf_{kn} \log \frac{N - df_k}{df_k}$. Moreover, to avoid the case of $\log \infty$, the formula can be corrected using $\frac{N - df_k + 0.5}{df_k + 0.5}$.

The rationale behind the $tf \times idf$ weighting scheme is that the degree to which a document n is relevant to a topic represented by term k is inversely proportional to df_k . If tf_{kn} is present, such degree is also directly proportional to the number of k term occurrences within document n .

Differently from classical indexing algorithms, in the OFAHIR methodology associations from terms to documents are taken into account and weights are assigned to them by computing a $tf \times idf$ -like weighting function that we call " $tf \times itf$ " weighting function. Given a term k , the $tf \times itf$ weighting function computes $tf_{kn} \log \frac{T}{itf_n}$ for each linked document n and given the number T of terms. The weight is inversely correlated to the degree of document exhaustivity, and directly correlated to the extent to which the topic represented by the term k is addressed in document n .

4.2 Querying

Querying takes as input both the description of documents and the description of a user's query. We consider natural language-based queries, i.e. sentences or string of terms without any predefined structure. According to the vector-space model, a query q is described as a vector $\vec{q} = (q_1, \dots, q_K)$, where K and descriptor $q_k = 1$ if term k occurs in q , 0 otherwise. A weighting function $w(q, k)$ can be defined by assigning query term weights to each term k . Weighting function $w(q, k)$ can be defined in several ways (Salton and Buckley 1987). As no particular assumptions on query are given, we choose a binary weighting function, i.e. $w(q, k) = q_k$.

Figure 6 Clustering in IR vs. clustering in OFAHIR

4.3 Clustering

The OFAHIR methodology is based on the clustering of information objects, such as documents and index terms to be presented to the user within an integrated environment based on browsing and querying processes. The clustering technique employed in OFAHIR aims to associate an information object with clusters of semantically related information objects.

The use of clustering within OFAHIR is quite different from the one within IR. Within IR, document clusters are employed as surrogates of documents during retrieval, and term clusters are used to add similar terms to expand the query. As stressed in Section 3 effectiveness of clustering depends on some parameters, such as the size of the document description that are out of user's control, and the user is unable to directly select which index terms employed in clustering do describe best his own informative need.

Within OFAHIR clustering is applied both on document and terms to make available clusters in hypertext form to be used for **browsing** mainly, i.e. the user is enabled to interactively navigate the hypertext by looking for terms and documents which describe best his own informative needs. Nodes of these hypertext clusters are either documents or terms. Automatically detected links connect the different types of nodes: document to similar documents, term to similar terms, document to index terms, and term to relevant documents.

Clustering is performed on the description of the information objects to be grouped. The clustering technique is based on **similarity functions** defined on the object descriptions, and working as association measures. Within the underlying vector-space model, the similarity function between two information objects is based on the angle between the corresponding vectors (Salton and McGill 1983). However, similarity function can be generalised to the probabilistic model once relevance data are incorporated into the weighting scheme (Croft and Harper 1979).

As similarity functions returns a real value, **ranking** can be applied on cluster elements. Values are normalised to the range $[0, 1]$. Clusters are created as follows. Given a query q submitted by the user, the associated cluster of relevant documents includes all documents n such that the following similarity measure:

$$sim(n, q) = \frac{\sum_{k=1}^K n_k q_k w(n, k) w(q, k)}{\sqrt{\sum_{i=k}^K n_k w(n, k)^2 \sum_{k=1}^K q_k w(q, k)^2}}$$

is over a stated threshold. For example, let $K = 3$, $N = 5$, and $\vec{q} = (1, 0, 1)$ be the number of index terms, the number of documents, and the description of query q . Let

$$W = \begin{bmatrix} 0 & 1.0 & 1.5 \\ 2.5 & 1.5 & 0 \\ 1.0 & 0 & 3.1 \\ 0 & 0 & 1.8 \\ 2.2 & 0 & 1.8 \end{bmatrix}$$

be a 5×3 matrix representing the weights of each index term within each document such that $W_{ij} = tf_{ij} \times idf_j$ is the weight of j th term within the i th document. For sake of simplicity, let assume that $w(q, k) = 1$ for each

term. By computing the scalar product between each row vector of matrix W and the vector $\vec{q}$, we get a vector (1.5, 2.5, 4.1, 1.8, 4) of similarity values corresponding the N documents.

Other than being judged as relevant to the query, retrieved documents can also work as entry points to browsing. This is the mean by which browsing and querying are integrated since from a retrieved document the user could select either the cluster of similar ones, or the cluster of terms describing the document content. Given a document n , the associated cluster of terms includes all terms k such that the following similarity measure:

$$\frac{w(n, k)}{\max_k w(n, k)}$$

is over a stated threshold, where $w(n, k) = tf_{kn} \log \frac{N}{df_k}$. The associated cluster of similar documents includes all documents m such that the following similarity measure:

$$\frac{\sum_{k=1}^K n_k m_k w(n, k) w(m, k)}{\sqrt{\sum_{k=1}^K n_k w(n, k)^2 \sum_{k=1}^K m_k w(m, k)^2}}$$

is over a stated threshold. The document similarity values are obtained by computing the matrix WW^T , where W^T is transpose matrix of W .

Given a term k , the associated cluster of documents includes all documents n that have the value of the following measure:

$$\frac{w(k, n)}{\max_n w(k, n)}$$

over a stated threshold, where $w(k, n) = tf_{kn} \log \frac{T}{idf_n}$, and tf_n is the number of index terms describing document n . The associated cluster of similar terms includes all documents h such that the following similarity measure:

$$\frac{\sum_{n=1}^N k_n h_n w(k, n) w(h, n)}{\sqrt{\sum_{n=1}^N k_n w(k, n)^2 \sum_{n=1}^N h_n w(h, n)^2}}$$

is over a stated threshold. The document similarity values are obtained by computing the matrix $W^T W$.

4.4 Browsing

In Section 3 ("Related Issues") we have already explained that clusters of index terms associated to each document can improve effectiveness. For ex-

ample, a cluster of similar documents can be linked to each document as well, and in this way a cluster includes all the documents considered as relevant to the topics addressed by the document which they have been associated to. This is a mean of increasing the quantity of retrieved information that is relevant to the user's information need. To present these kinds of relationships to users is a way of supporting browsing. As clusters are presented in a hypertext form, browsing means moving from a starting node to an ending one. Whenever the user selects a node a hypertext cluster needs to be constructed "on-the-fly". The nodes of these hypertext clusters represent the objects which are associated to the object represented by a starting node. The user selects the ending node by choosing one object from the cluster. The interaction between browsing and clustering is presented in Figure 7.

Figure 7 The interaction between browsing and clustering processes within the model of the OFAHIR system

To support browsing, four types of link have been identified. These types are presented in the following.

Referring links allow the user to retrieve the documents that are about the same subject of the document he is looking at. The similar documents are ranked and presented to the user accordingly to a measure of aboutness. Document subject is represented by occurring index terms. Since this measure is function of the number of co-occurring index terms and of the $tf \times idf$ -based weight, top-ranked documents are the ones having high num-

bers of common index terms, and high index term weights. High index term weights discriminate the documents that are more relevant to the subject represented by the common index terms. By choosing the most similar documents, the user selects the ones that describe that subject more extensively and specifically than the others. For short, these links are depicted as DD links in Figure 7.

Description links allow the user to retrieve the index terms that describe the different subject aspects of the document he is looking at. The index terms are ranked and presented to the user accordingly to the $tf \times idf$ -based weight. Top-ranked index terms represent the subject aspects that are treated within the other documents with a high degree of specificity. By choosing the most weighted index terms, the user selects the ones that allow him to further retrieve the documents that are most relevant to the specific subject aspects represented by the selected index terms. The description links are named DT links in Figure 7.

Dictionary links allow the user to retrieve the index terms that are “synonyms” of the index term he is looking at. The similar index terms are ranked and presented to the user accordingly to a measure of “synonymy”. Synonymy is measured by the number of co-occurring documents and by the $tf \times itf$ -based weight. Top-ranked index terms are the ones having high numbers of common documents, and high index term weights. Given an index term t , high weights discriminate the similar index terms that more specifically represent the subject represented by t . By choosing the most similar index terms, the user selects the ones that allow him to further retrieve the documents that are most relevant to the represented subject aspects. These links are named TT links in Figure 7.

Context links allow the user to retrieve the documents that are about the subject represented by the index term he is looking at. The documents are ranked and presented to the user accordingly to the $tf \times itf$ -based weight. Given an index term t , high weights discriminate the documents that deal with the subject represented by t more specifically than the ones having a lower rank. These context links are named TD links in Figure 7.

Links between index terms and documents are obtained starting from the term-document occurrence information. The links between index terms and documents (TD links) and the symmetric ones (DT links) are thus static since they are inserted once for all at indexing-time, even though new index terms, and then documents can be added. The basic hypertext is then the one including links between index terms and documents only. Browsing allows the user to access nodes representing documents and index terms by following DT or TD links. Links between index terms (TT links) and those between documents (DD links) are automatically built at browsing-time, i.e. whenever the user asks to access an index term node from another index term, or to access a document node from another document node. The DD and TT links are

built through a **clustering** process which groups together similar information objects of the same level, i.e. documents or index terms. Through clustering, the user can access, for example, the set of documents that are similar to a given document. The reason why the DD and TT links are built “on-the-fly”, i.e. at search-time is that the re-computation of links is avoided whenever the insertion of new documents are inserted, and thus new index terms are created. The insertion of an additional document requires the indexing of the document itself, the creation of the nodes corresponding to the new index terms, and the construction of the links between the new document and the index terms.

Figure 8 The hypertext structure

5 HYPERTEXT STRUCTURE

Figure 8 depicts the general structure of a hypertext that results from the process of applying the OFAHIR methodology to an initial flat document collection. The OFAHIR methodology generates an automatically authored hypertext which is built making use of the types of links that have been described in Section 4.

The hypertext that results from the application of the OFAHIR methodology is organised in two distinct levels: the document level, and the index term level. Characteristics of the items of these two levels are presented in the following of the section.

- **Document level** includes three types of node. Each of them correspond to three different ways of viewing the semantics of a given document:
 1. the **full-text** of the document together with the links to the cluster of similar documents, and to the cluster of index terms, as described below;
 2. the **cluster of similar documents** implemented as a HTML page storing the list of links to the full-text of the documents which have been considered as similar to the previous one;
 3. the **cluster of index terms** implemented as a HTML page storing the list of links to HTML pages stored on the index term level.

The semantics of a document is then accomplished by the original full-text, by the cluster of documents that have been considered as relevant to the same topic, and by the cluster of index terms the system employ to describe the document semantics. Similarity values can help the user choose most relevant document or most similar index term.

- **Index term level** includes just one type of node storing two clusters: the former is implemented as the list of links to the similar index terms, the latter corresponds to the list of full-text documents considered as relevant to the topic expressed by the index term.

The semantics of an index term is thus carried out by two clusters, other than by the string of words or letter forming the term itself. The cluster of similar terms provides the user with information about the way the system relates the term to the others. By means of the cluster of similar terms the user can directly select, for instance, synonymous terms, or more general terms. The cluster of documents play a twofold role. Firstly, the user can effectively retrieve the documents that are relevant to the topic; as documents are presented to the user as ranked by a similarity, the user has knowledge about most relevant ones. Secondly, the user can “move” himself to another document that address the topic to a greater or lesser degree according to the ranking similarity value.

6 A TOOL FOR OFAHIR

A new experimental system environment has been designed and developed to implement the OFAHIR methodology that has been presented in Section 4. An overview of this system environment and its architecture are given in Figure 9 where are also depicted most relevant components.

Each component has been designed and implemented to solve a specific task or a set of similar tasks. The components of the architecture are modular, and each of them can be also re-used in a different object-oriented context.

The system can work in a World Wide Web (WWW or Web) environment since the components performing the data retrieval have been implemented

Figure 9 The overall architecture of the system implementing OFAHIR

according to the CGI protocol. This means that through any available Web client a user can pose queries to and/or browse the database that manages the documents collection and the index terms structure. Information objects have been implemented as HTML pages, and hypertext links have been achieved using HTML anchors.

The system has six main components, that are sketched in Figure 9. The components that are sketched in white are the four that are used to input the collection of documents, and prepare the basic objects that are needed at run time to build on the fly the clusters. The clusters are used to answer the querying and browsing actions of the user of the system. Those components are: **EXODUS**, the **IR object class library**, the **IR Object Builder (IROB)**, and **DBTachir**. The two components that are sketched in grey are those ones that make possible the interaction between user and system, that is: the **Querying Tool (QT)**, and **On the Fly Author of Hypertexts for IR (OFAHIR component)**. In the remaining part of the section, most relevant characteristics of each component are presented.

- **EXODUS** is an object storage manager developed by the University of Wisconsin in the context of the EXODUS project. A general presentation and an overview of the project can be found in (Carey, DeWitt and Vandenburg 1988). A characteristic of EXODUS that is relevant to the development and management of hypertext and information retrieval applications is that it facilitates the fast development of high-performance databases to be used for specific applications. EXODUS is the component

to be used as storage manager of the object database that maintains the collection of relevant objects for associative retrieval.

- the **IR object class library** is the library of classes that implement the fundamental operations and structures of a hypertext information retrieval system. The library is currently implemented in E, the C++ based language of the Exodus project, this language allows for objects persistence (Richardson, Carey and South 1993). Figure 10 depicts a significant part of the scheme of the IR object class library.

Figure 10 The IR object class library: a relevant excerpt

- the **IR Object Builder (IROB)** performs indexing and populates the object database; two different kinds of inputs are used by IROB: the **collection of documents** that has been chosen to be managed by the application and the **list of words** that are not semantically meaningful for the used collection of documents, that is the list of words that is identified in the context of these kinds of applications as “stop list” or “black list”, because it contains the words that are not retained as significant during the indexing process (see, for example, (Salton and McGill 1983)). The collection of documents is prepared in the form of a set of files that store the original documents in HTML format. IROB makes use of the IR object class library for producing as output the persistent objects that are stored in an object database instance that is managed by EXODUS. Indexing can be incremental, this means that new documents can be added to the original document collection and they can be given as input to IROB which

incrementally updates the object database; the computation of document-term associations is made only for those new documents, together with the assignment to index terms.

- **DBTachir** interface between the storage manager and the OFAHIR applications by making objects instantiated from the class library persistent.
- The **Querying Tool (QT)** implements the querying process, so it is the system component that together the browsing facility supports the retrieval of the information for the final user. QT takes as input the user query, that the user can formulate making use of a HTML form; QT processes the query and produces as output the HTML page storing a list of links to HTML pages. Such a list of links is the cluster of documents that are judged by QT as relevant to the query. Since HTML pages are linked to other list of links, the QT result can work as entry point to browsing, and then as input of the OFAHIR component. The list of links produced by QT contains the ordered list of anchors to documents that have been computed as relevant to the query by QT; the order is descendent in relation with decreasing weigh of pertinence of the document to the user query.

Figure 11 OFAHIR clustering using HTML primitives

- The **OFAHIR component** automatically constructs the hypertext clusters to be used as supporting tools for browsing through the object database. This component takes as input a link occurring within a HTML page, and gives as output a HTML page including a list of links to HTML pages which represent the associated information objects. The HTML page stor-

ing the list of links corresponds to the cluster of associated objects. Such a correspondence is depicted in Figure 11. The list of links is ranked by a similarity value. The cluster of selected objects is interactively presented to the user using Web hypertext primitives. This design choice has made possible the interaction between the system and the final user since the user can formulate the queries through a Web client. The OFAHIR component interacting with the other components of the environment produces dynamically the HTML pages that the user can see by means of a Web browser.

7 EVALUATION AND PERFORMANCE

7.1 Evaluation

Classical evaluation measures used in IR are **precision** and **recall**. Precision is the proportion of relevant documents given a set of retrieved documents, recall is the proportion of retrieved documents given a set of relevant documents. Precision and recall have been proposed and are still employed to measure the effectiveness of a batch IR system in retrieving information. Thus the evaluation of automatically constructed hypertexts for IR becomes a very complex issue since the classical evaluation measures are only partially usable in the evaluation of characteristics of hypertext information retrieval systems. The intrinsic interactivity of automatically constructed IR hypertexts, i.e. the active role and direct use of the user make the evaluation process based on precision and recall measures very difficult. It is therefore necessary to develop new methods that establish a relationship between present evaluation efforts to previous evaluation work in information retrieval. To establish and keep this relationship permits the maintenance of the possibility of comparing experimental results reached in the past and those reached with present systems. Together with keeping the possibility of comparison among experimental data, a new proposed method should in general evaluate effectively new system capabilities, and in particular address the evaluation of those applications and systems that have been automatically developed and made available to the users.

Some efforts in the evaluation of automatically constructed IR hypertexts have been recently conducted. The research work in evaluation can be broadly classified into two types:

1. The evaluation of IR hypertext effectiveness from the **user-side** by carrying out operational or laboratory tests with the active and direct users' participation. For instance in (Dunlop 1991, Dunlop and van Rijsbergen 1993) a model of hypermedia IR has been proposed and tested with real users by

- carrying out two experiments that make use of a text document collection to relate the results to previous findings in the information retrieval area.
2. The evaluation of IR hypertext effectiveness from the **system-side** by computing quantitative measures by simulating retrieval processes, such as querying and browsing through the use of some test collections; this type of evaluation can consist of:

- The classical precision-recall computation. In (Croft and Turtle 1993), for example, a comparison of performance of the strategies used in two retrieval models is made; a probabilistic retrieval model incorporating inter-document links with strategies that ignore the links versus a heuristic spreading activation strategy is evaluated.
- The computation of measures describing the hypertext topology. An interesting approach is presented in (Furner, Ellis and Willett 1996) where the authors report on an experimental study of comparison of hypertexts, where different people were asked to produce a hypertext representation of the same full-text document. The investigation has required the calculation of measures of similarity between pairs of manually produced hypertexts. It is showed that the structure of a hypertext document is crucially dependent upon the person who has 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 only a specific designing and development view, but also it is reproducible starting from the same initial collection of flat documents.

We have designed and carried out some experiments to test the effectiveness in using a hypertext automatically built by the OFAHIR tool. The tests we have designed and conducted are of the system-side type and they adhere to the classical precision-recall computations previously described. We have decided to simulate some browsing paths as they were performed by a user, and we have compared the results with those paths that were obtained by a query-based retrieval process to test if the use of browsing can help in improving the effectiveness with respect to the query-based retrieval.

Each test starts with a query-based retrieval to build an initial ranked document list. After the query-based retrieval, links are followed by starting either from the query terms or from the retrieved documents. The ending of each browsing path is another list of retrieved documents. The first half of the latest retrieved document list replaces the first half of the initial retrieved document list to build a new list, as depicted in Figure 12. The rationale behind this simulation is the comparison between two *tactics* the user can perform in using an OFAHIR hypertext: the use can either (1) scan the whole list of initially retrieved documents, or (2) stop the scanning after having viewed half

Figure 12 Merging the browsing-path-based retrieval results to the query-based retrieval results.

document list and try to search more documents by browsing. The first half of the document list reached through browsing is scanned in place of the second half of the initial document list. The total number of examined documents is then the same, but in the second tactic the user follows the hypertext links instead of examining the whole initial document list. The retrieval effectiveness has been measured through the classical recall and precision ratios computed for different groups of retrieved documents, i.e. groups of 5, 10, 15, 20, 25, 30, 35, and 40 documents. Recall and precision ratios are computed for every retrieved document list, and thus different values are compared for each tactic and browsing path. In particular, two browsing paths has been tested to test two specific types of links:

1. The **dictionary links** inserted between the index terms and based on term similarity. The browsing path consists of the selection of up to four query terms from which the dictionary links are computed to connect a list of similar terms, i.e. a list for each query term. Query terms are ranked according to their own $tf \times idf$ weight, and the first selected query term is the most weighted one, and so on. The list of similar terms is ranked by the similarity measure as illustrated in Section 4.3. From each similar term, a list of related documents is built and ranked by the $tf \times idf$ weight. The first half of each list replaces the second half of the initial document list. The recall and precision values computed for groups of 5, 10, 15, and 20 documents retrieved through browsing are respectively compared to

	5	10	15	20	25	30	35	40
Recall	0.414	0.542	0.625	0.669	0.704	0.741	0.782	0.819
Precision	0.291	0.211	0.175	0.146	0.125	0.108	0.099	0.091

Table 1 ADI test collection. Results of the query-based retrieval.

the values computed for groups of 25, 30, 35, and 40 documents retrieved through the query.

2. The **referring links** inserted between the documents and based on document similarity. The browsing path consists of the selection of up to four retrieved documents from which the referring links are computed to connect a list of similar documents, i.e. a list for each retrieved document. The list of similar documents is ranked by the similarity measure as illustrated in Section 4.3. The first half of each list of similar documents replaces the second half of the initial document list. The recall and precision values computed at 5, 10, 15, and 20 documents retrieved through browsing are respectively compared to the values computed at 25, 30, 35, and 40 documents retrieved through the query.

In IR, test collections are employed to test and compare models and methods. Each test collection consists of a set of documents, a set of queries, and a list of judgements about the relevance of each document with respect to each query. Two test collections are employed to perform our experiments: the ADI collection, a very small collection of 83 documents and 35 queries, covering the IR area, and the Cranfield collection, consisting of 1400 documents and 225 queries on aeronautics, that is being used in IR for a long time. The size of these test collections is unrealistic if compared to the real IR application and to the most recent test collections used in the TREC (Text REtrieval Conference, (Harman 1995)). However, at this stage of the work, we are interested on the mechanisms underlying the effectiveness of the OFAHIR links, and we believe that these collections are large enough for this purpose.

Some preliminary results of the experiments using the ADI collection are reported in Tables 1, 2, and 3, and the results for the Cranfield collection are reported in Tables 4, 5, and 6. Tables 1 and 4 report the precision and recall values for the query-based retrieval for the ADI and the Cranfield collections respectively. Tables 2, 3, 5, and 6 report the differences in percentage between the precision and recall values obtained using the two browsing path for the two collections. Tables 2 and 5 report on the retrieval effectiveness of the browsing path through referring links starting from four retrieved documents. Tables 3 and 6 report on the retrieval effectiveness of the browsing path starting from the fourth ranked query term, through the dictionary links

Document	25	30	35	40	Average
1	4.0%	1.4%	-1.4%	-3.2%	0.2%
	5.5%	7.1%	5.8%	3.1%	5.4%
2	1.0%	-0.6%	-2.9%	-1.1%	-0.9%
	1.8%	4.4%	1.7%	-0.8%	1.8%
3	1.2%	-0.4%	-1.5%	-2.2%	-0.7%
	0.0%	1.8%	-0.0%	-0.8%	0.2%
4	-3.1%	-4.7%	-7.5%	-8.5%	-5.9%
	-2.8%	0.0%	-3.3%	-2.3%	-2.1%

Table 2 ADI test collection. Results of the tactic based on document similarity links. Recall (first row) and precision (second row) are computed after 25, 30, 35, and 40 documents starting from four different documents.

	25	30	35	40	Average
from 4th query term					
<i>1st term</i>	3.7%	3.8%	2.3%	-0.3%	2.4%
	3.7%	4.4%	4.1%	3.9%	4.0%
<i>2nd term</i>	0.9%	-2.1%	-2.0%	-3.4%	-1.6%
	0.9%	0.0%	-1.7%	-3.1%	-1.0%
<i>3rd term</i>	1.3%	2.9%	-2.3%	-3.0%	-0.3%
	0.9%	3.5%	-1.7%	-1.6%	0.3%
<i>4th term</i>	0.1%	-1.8%	-5.1%	-7.0%	-3.4%
	0.9%	1.8%	-1.7%	-3.1%	-0.5%

Table 3 ADI test collection. Results of the tactic based on term similarity links. Recall (first row) and precision (second row) are computed after 25, 30, 35, and 40 documents starting from four different terms similar to a query term.

	5	10	15	20	25	30	35	40
Recall	0.002	0.002	0.003	0.006	0.006	0.009	0.010	0.011
Precision	0.006	0.005	0.004	0.004	0.003	0.004	0.004	0.003

Table 4 Cranfield test collection. Results of the query-based retrieval

document	25	30	35	40	Average
1	55.0%	11.7%	16.0%	24.5%	26.8%
	38.9%	7.7%	14.3%	22.6%	20.9%
2	7.1%	-23.4%	-15.1%	-11.8%	-10.8%
	11.1%	-7.7%	-3.6%	0.0%	-0.0%
3	-7.1%	-32.2%	-28.3%	-6.3%	-18.5%
	-11.1%	-26.9%	-21.4%	-16.1%	-18.9%
4	32.3%	27.4%	22.9%	37.5%	30.1%
	5.6%	-7.7%	3.6%	6.5%	2.0%

Table 5 Cranfield test collection. Results of the tactic based on document similarity links. Recall (first row) and precision (second row) are computed after 25, 30, 35, and 40 documents starting from four different documents.

from 4th query term	25	30	35	40	Average
<i>1st term</i>	43.4%	26.0%	24.1%	32.0%	31.4%
	22.2%	11.5%	21.4%	32.3%	21.9%
<i>2nd term</i>	8.2%	1.9%	8.8%	21.6%	10.1%
	0.0%	-15.4%	-3.6%	3.2%	-3.9%
<i>3rd term</i>	30.9%	12.5%	11.8%	19.5%	18.7%
	16.7%	-7.7%	0.0%	-0.0%	2.2%
<i>4th term</i>	27.0%	-5.1%	-14.1%	-3.4%	1.1%
	11.1%	-15.4%	-21.4%	-16.1%	-10.5%

Table 6 Cranfield test collection. Results of the tactic based on term similarity links. Recall (first row) and precision (second row) are computed after 25, 30, 35, and 40 documents starting from four different terms similar to a query term.

connecting four similar term, and ending to the corresponding list of related documents ranked by $tf \times itf$ weight.

Some comments can be drawn from these preliminary results. The magnitude of the differences between the precision and recall values is lower for the ADI collection than for the Cranfield collection. The effectiveness of the query-based retrieval for the ADI collection is higher than for the Cranfield collection. One might say that the effectiveness of browsing is as higher as less the query-based retrieval is effective. For the ADI collection, it seems that the use of referring links leads to a higher improvement in precision than in recall. However the use of referring links between documents or of dictionary

links between term lead to different conclusions dependently on the employed collection and further analysis are therefore necessary.

7.2 Performance

In spite of its effectiveness and extensive research, clustering has been only partially implemented within experimental systems, and rarely within commercial systems, even though some Web search engines, such as Excite provides with links to similar documents. One reason is that clustering algorithms, such as those in OFAHIR have to be performed within an interactive environment, i.e. at real-time. Even though the increase of computing performance, the design of efficient algorithms (Silverstein and Pedersen 1997) is a necessary condition to make clustering a feasible method for interactive environment. Clustering requires the computation of similarity measures between object pairs. If clustering was statically performed, all similarity measures should be computed at indexing-time. This is clearly infeasible whenever the number of objects is high, as well as it requires the re-computation of similarity measures whenever a new object is inserted. OFAHIR clustering is dynamically performed, i.e. at browsing or querying-time. The advantage of OFAHIR clustering is that no similarity re-computation is necessary whenever a new object is inserted. The construction of a cluster, such as a cluster of index terms associated to a given object is similar to the querying process since it requires the computation of a similarity value between the object and each cluster item. Some experiments have showed that the current implementation of OFAHIR clustering is still inefficient, and that more efficient algorithms are needed.

8 CONCLUSIONS AND FUTURE WORKS

We presented a new methodology, called OFAHIR, to manage and retrieve information stored in large collections by means of “on-the-fly” automatically authored hypertexts. Together with the methodology we have presented main characteristics of an experimental tool that implements the methodology itself. The results presented in the paper constitute a step further towards the definition of an innovative and complete associative retrieval model, that is the main target of the project of which OFAHIR is a part.

We plan to further develop the methodology integrating in it the use of other techniques, such as thesauri, classification systems, and conceptual networks. Techniques to interactively build query by browsing, i.e. to support the user’s query formulation in the selection of terms from the hypertext and to adapt link construction to a given user’s model are also under investigation. We are in particular investigating the possibility of incorporating in OFAHIR the

automatic construction of links that are built taking into active account also user characteristics.

Finally, we are carrying out further tests to evaluate the automatically built hypertexts with different collections and browsing paths. A user-centred evaluation is also under design and development.

9 ACKNOWLEDGMENTS

We would like to thank the anonymous referees that provided many useful comments to the initial version of this paper. This work has been partially supported by the 1996 MURST 40% project on "Advanced Database Systems". From February 1997 Lucio Benfante has been supported by a grant of the "Consorzio Padova Ricerche (CPR)" in the context of the "Semantic Networks Inter-Operation" project supported by the European Space Agency (ESA) and that sees the participation of VITROCISET Space Division together with CPR.

REFERENCES

- Agosti, M. and Allan, J. (eds): 1997, *Special issue on methods and tools for the automatic construction of hypertexts*, Vol. 33(2) of *Information Processing & Management*.
- Agosti, M., Crestani, F. and Melucci, M.: 1996, Design and implementation of a tool for the automatic construction of hypertexts for Information Retrieval, *Information Processing & Management* **32**(4), 459–476.
- Agosti, M., Crestani, F. and Melucci, M.: 1997, On the use of information retrieval techniques for the automatic construction of hypertexts, *Information Processing & Management* **33**(2), 133–144.
- Agosti, M. and Smeaton, A. (eds): 1996, *Information retrieval and hypertext*, Kluwer Academic Publishers, Boston, MA, USA.
- Belkin, N., Oddy, R. and Brooks, H.: 1992, ASK for information retrieval. part i. background and theory, *Journal of Documentation* **38**, 61–71.
- Brusilovsky, P.: 1996, Methods and techniques of adaptive hypermedia, *User Modeling and User Adapted Interaction* **6**(2-3), 87–129.
- Carey, M. J., DeWitt, D. J. and Vandenburg, S. L.: 1988, A data model and query language for EXODUS, *Proceedings of the Special Interest Group on Management of Data (SIGMOD) Conference*, ACM, Chicago, IL, pp. 413–423.
- Croft, W. and Harper, D.: 1979, Using probabilistic models of document retrieval without relevance information, *Journal of Documentation* **35**, 285–295.
- Croft, W. and Turtle, H.: 1993, Retrieval strategies for hypertext, *Information Processing & Management* **29**(3), 313–324.

- Dunlop, M.: 1991, *Multimedia Information Retrieval*, PhD Thesis, Department of Computing Science, University of Glasgow, Glasgow, UK.
- Dunlop, M. and van Rijsbergen, C.: 1993, Hypermedia and free text retrieval, *Information Processing & Management* **29**(3), 287–298.
- Furner, J., Ellis, D. and Willett, P.: 1996, The representation and comparison of hypertext structures using graphs, in M. Agosti and A. Smeaton (eds), *Information retrieval and hypertext*, Kluwer Academic, pp. 75–96.
- Griffiths, A., Luckhursts, H. and Willett, P.: 1986, Using inter-document similarity information in document retrieval systems, *Journal of the American Society for Information Science* **37**, 3–11.
- Harman, D.: 1988, Towards interactive query expansion, in Y. Chiaramella (ed.), *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, Grenoble (France), pp. 321–332.
- Harman, D.: 1992, Evaluation issues in Information Retrieval, *Information Processing & Management* **28**(4).
- Harman, D.: 1995, The TREC conferences, in R. Kuhlen and M. Rittgerger (eds), *Hypertext-Information Retrieval-Multimedia (HIM)*, Konstanz, Germany, pp. 9–28.
- Hearst, M. and Karadi, C.: 1997, 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, PA, USA, pp. 246–256.
- Ingwersen, P.: 1992, *Information Retrieval Interaction*, Taylor Graham, London.
- Kwok, K.: 1996, A new method of weighting query terms for ad-hoc retrieval, in H.-P. Frei, D. Harman, P. Schäuble and R. Wilkinson (eds), *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, ACM Press, New York, pp. 187–196.
- Lesk, M.: 1997, *Practical Digital Libraries: Books, Bytes, and Bucks*, Morgan Kaufmann, San Francisco, CA, USA.
- Maron, M. and Kuhns, J.: 1960, On relevance, probabilistic indexing and retrieval, *Journal of the ACM* **7**, 216–244.
- Richardson, J., Carey, M. and South, D.: 1993, The design of the E programming language, *ACM Transactions on Programming Languages and Systems* **15**(3), 494–534.
- Rocchio, J. J.: 1971, Relevance feedback in information retrieval, in G. Salton (ed.), *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall, Englewood Cliffs, NJ, chapter 14, pp. 313–323.
- Salton, G.: 1963, Associative document retrieval techniques using biblio-

- graphic information, *Journal of the ACM* **10**, 440–457.
- Salton, G.: 1968, *Automatic information organization and retrieval*, Mc Graw Hill, New York.
- Salton, G. and Buckley, C.: 1987, Term weighting approaches in automatic text retrieval, *Information Processing & Management* **24**(5), 513–523.
- Salton, G. and Buckley, C.: 1990, Improving retrieval performance by relevance feedback, *Journal of the American Society for Information Science* **41**(4), 288–297.
- Salton, G. and McGill, M.: 1983, *Introduction to modern Information Retrieval*, McGraw-Hill, New York.
- Saracevic, T., Spink, A. and Wu, M.: 1997, Users and intermediaries in information retrieval: What are they talking about?, in C. P. Anthony Jameson and C. Tasso (eds), *Sixth International Conference on User Modeling*, Springer Wien, New York, Chia Laguna, Sardegna, Italy, pp. 43–54.
- Silverstein, C. and Pedersen, J.: 1997, Almost-constant-time clustering of arbitrary corpus subsets, *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, Philadelphia, PA, USA, pp. 60–66.
- Sparck Jones, K.: 1971, *Automatic keyword classification*, Butterworths.
- Sparck Jones, K.: 1972, A statistical interpretation of term specificity and its application in retrieval, *Journal of Documentation* **28**(1), 11–21.
- Sparck Jones, K.: 1979, Experiments in relevance weighting of search terms, *Information Processing & Management* **15**(3), 133–144.
- van Rijsbergen, C.: 1979, *Information Retrieval*, second edn, Butterworths, London.
- Willett, P.: 1988, Recent trends in hierarchic document clustering: a critical review, *Information Processing & Management* **24**(5), 577–597.

10 BIOGRAPHY

MARISTELLA AGOSTI is Associate Professor of Computer Science in the Department of Electronics and Informatics (DEI) of the University of Padua, Italy, and group leader of the Information Management Systems (IMS) research group of the Department. Main research interests: object-oriented environments for the management of persistent information; methods and tools for the automatic construction of hypertexts; networked and multimedia information management; digital libraries; evaluation of interactive retrieval systems. She has authored more than 100 papers published in scientific journals, books, and conference proceedings. She has co-edited the book on "Information Retrieval and Hypertext" (Kluwer Academic, 1996) together with A.F. Smeaton. Advisor for the evaluation and reviewer of EU proposals and projects under ESPRIT, INFO2000, Information Engineering and Libraries programmes. Member of the editorial boards of *Information Processing &*

Management, and *The Computer Journal*. Subject Area Editor (Hypermedia) of *Information Retrieval*. She has been member of the program committees of international conferences, such as ACM SIGIR, ACM Digital Libraries, EDBT, and COLIN. She is member of the ACM and IEEE Computer Society.

LUCIO BENFANTE received a *Laurea* degree in Statistics from the University of Padua, Italy, in 1996, with a thesis in computer science. From February 1997 he is a grant holder of the "Consorzio Padova Ricerche (CPR)" in the context of the "Semantic Networks Inter-Operation" project supported by the European Space Agency (ESA). Areas of research and experience: object-oriented and relational database management systems; automatic indexing for information retrieval; construction and persistent management of conceptual networks; automatic construction of HTML pages. From 1993 to 1996 he has participated in the development of professional data management software. In 1993 he has collaborated with the JCE publisher for fractals software development.

MASSIMO MELUCCI received a *Laurea* degree in Statistics from the University of Padua, Italy, in 1990 and a PhD degree in Computing Science in 1996. Since 1991 he has been working with the Information Management System research group of the Department of Electronics and Informatics of the University of Padua in the area of object-oriented modelling of Information Retrieval (IR) applications, passage retrieval, and automatic construction of hypertexts. He has participated in national and international research projects. He currently is a post-doctoral fellow at the University of Padua. His current main research interests are IR, evaluation of automatically constructed hypertexts, digital libraries, multi-lingual IR, passage retrieval.