Kybernetika

Gloria Bordogna; Gabriella Pasi

Flexible representation and querying of heterogeneous structured documents

Kybernetika, Vol. 36 (2000), No. 6, [617]--633

Persistent URL: http://dml.cz/dmlcz/135377

Terms of use:

© Institute of Information Theory and Automation AS CR, 2000

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.

This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* http://project.dml.cz

FLEXIBLE REPRESENTATION AND QUERYING OF HETEROGENEOUS STRUCTURED DOCUMENTS

GLORIA BORDOGNA AND GABRIELLA PASI

In this paper we present a fuzzy model for representing documents having a hierarchical structure and possibly containing multimedia information. We consider an archive containing documents with distinct (heterogeneous) logical structures. We also propose a flexible query language for expressing soft selection conditions on the structured documents. The documents' content is organized into thematic (topical) sections where the index terms play a distinct role. The proposed document representation is adaptive to the user, who can indicate the preferred sections of documents, i. e. those which they estimate to bear the most interesting information, and can linguistically quantify the number of sections which determine the global potential interest of the documents. Linguistic quantifiers in the query specify the approximate number of the sections in which the query terms should appear.

1. INTRODUCTION

In several applications of information retrieval, the considered archives contain documents where the information is organized into a structure, which can be aimed either at specifying the distinct semantics of the documents sections or at organizing the appearance of the text in a graphical format. In the former case we can talk about a "logical" structure of the document, which associates a distinct informative role with distinct pieces of information. Let us think about the structure of a scientific paper, which is organized in sections such as *Title, Authors, Introduction, References.* Moreover a distinct informative role is determined by distinct information media, such as text, images, and sound. For example, this kind of structure can be explicitly defined by means of the XML language.

The latter case refers to the organization of a text in a "typographical" appearance: for example, different dimensions of characters and underlined words are used in order to put a distinct emphasis on distinct documents subparts. This structure is supported by standards such as the HTML language.

Most of the existing Information Retrieval Systems (IRSs) apply document indexing that does not take into account the documents structure; moreover, the indexing procedure can be seen as a black box producing the same document representation for all users. Of course this is a simplification adopted to improve the systems effi-

ciency, but at the expenses of the systems effectiveness. In fact, when examining the content of a document a user would naturally privilege some subparts of the documents structure, depending on his/her interest. With this "rigid" representation, the user cannot interact with the system with the aim of influencing the representation of the documents' content.
It is well known that the quality of the user-system interaction greatly influences the effectiveness of the query evaluation results [24, 27]. However, in IRSs the user-system interaction is only at the level of query specification and relevance feedback. When formulating requests, users are forced to translate their information needs, often pervaded by vagüeness, into a formal query such as a Boolean query, that only makes possible the specification of crisp selection conditions. This simplification constitutes a severe limitation since users would naturally formulate vague requirements for expressing soft selection conditions, i. e., conditions that can be satisfied to a gradual extent. For this reason, several approaches have been proposed in the literature to define flexible query languages and partial matching mechanisms which are tolerant to imprecision in the phases of query formulation and interpretation [1, 2, 4, 6, 12, 17, 24, 27]. In these approaches, the first aim is to design IRSs able to effectively rank the retrieved documents in decreasing order of their estimated relevance or probability of relevance to the user needs. These approaches share the point of view that the more the query language and retrieval mechanism are flexible so as to allows the faithful expression and interpretation of user needs, the more the effectiveness of IRSs can be improved.
Due to both the great amount of multimedia and hypermedia information widely available on wide-area networks, and to the diffusion of the de-facto standards for structured documents such as SGML, HTML and XML documents, there is a growing need for new conceptual models for representing and querying structured documents [9, 13, 14, 15].

In this paper we propose a fuzzy model for representing documents having a logical structure, that in the most general case can be represented by a graph. The considered documents may contain multimedia information with different (heterogeneous) structures. A further assumption is that the usefulness of sections to the users as potential carriers of relevant information varies depending on users needs. For this reason we also propose a flexible query language for expressing soft content-based selection conditions on the structured documents. The motivation for defining a fuzzy representation of documents is that fuzzy set theory provides a natural framework to manage vague and subjective concepts [30]. The proposed document representation is adaptive in the sense that it can be tuned by the users according to their search interests. To this aim, in a query, they have to indicate the preferred sections of documents, i. e. those which they estimate bearing the most interesting information. Further users can linguistically quantify the number of sections which determine the global potential interest of the documents through the specification of a linguistic quantifier such as *all, most of, at least 70% of*, etc. expressing stricts or more relaxed constraints [28, 29, 31]. In this way, users can tune the view of the document content used by the retrieval mechanism to determine the document relevance.

In the following section a review of the flexible indexing models for IRSs defined in the literature is presented, then in Section 3 the fuzzy representation of structured documents is described. In Section 4 the formalization of the retrieval function of structured documents against queries is presented. Finally, the conclusion section summarizes the achieved results.

2. FLEXIBLE INFORMATION RETRIEVAL SYSTEMS

The expression "flexible IRSs" refers to IRSs that are adaptive to users who are interacting with them. Generally, flexibility has been pursued by defining retrieval models that tune their behavior by applying some relevance feedback mechanism. These approaches try to focus users' needs by analysing the documents considered the most relevant by the user in response to a query [24, 27].

The problem of flexibility in IRSs has also been approached by defining flexible query languages [17, 21, 22, 23, 24]. In particular, since the Boolean query language has been the most adopted one and at present is still very common for example in libraries and in search engines on the web, some research efforts have been spent to extend it with the aim of simplifying and enhance its expressiveness. This has been pursued at a twofold level: on the one hand more powerful selection conditions have been defined, and on the other hand soft aggregation operators for combining the selection conditions have been adopted. The former has been tackled by allowing the association with each query term (search term) of an indication of terms importance [1, 2]. The latter aims at defining operators which offer a trade-off between the AND (requiring the presence of all the search terms), and the OR (requiring the presence of at least one of the search terms) [17, 23]. The adoption of softer aggregation operators can avoid the rejection of useful items as a result of too restrictive queries, and the retrieval of useless material in reply to general queries. Among the fuzzy extensions of the Boolean query language, some are aimed at introducing linguistic elements for the purpose of capturing the vagueness of the user needs as well as to simplify the user-system interaction [2, 4, 12]. This has been faced at two levels:

- through the definition of more expressive as well as softer selection criteria, which allow the specification of the distinct importance of the search terms to the users needs [2, 12].

- through the simplification of the Boolean structure of the query by introducing soft connectives of the selection criteria, characterized by a parametric behaviour which can be set between the two extremes AND and OR. In [4] the Boolean query language has been generalized by defining aggregation operators that are specified by linguistic quantifiers such *as at least k or about k.*

The common basis of flexible IRSs is the adoption of a weighted representations of documents. This makes possible the definition of a partial matching mechanism able to estimate either the relevance of each retrieved document to a gradual extent [24, 25, 26] or the probability of relevance of documents [27]. Another possibility for achieving flexibility in an IRS is to incorporate user adaptive indexing mechanisms. There is a need for new, more flexible measures of representativity of terms that allow for different ranking of documents according to user or query types [5, 13, 19]. To the aim of indexing multimedia structured documents, a model that is based on views reflecting the potential ways of "seeing" multimedia documents has been defined in [19].

By sharing the idea that the semantic content of documents reflects one way of seeing and using them, we propose in this paper a fuzzy representation of structured documents that can be adapted to the user.

3. A FUZZY REPRESENTATION OF STRUCTURED DOCUMENTS

In many application fields, it often happens that one has to generate representations of documents of heterogeneous collections which are characterized by different logical structures. For example, the scientific papers published in journals or series of volumes can be structured in several ways. In order to clearly organize the paper content, the author may decide to structure the paper in a given number of sections and subsections; on the oder side he/she must adhere to the style and organization imposed by the journals or book series publisher who may require to start the paper with a title section, then with the authors name and affiliation, an abstract or a set of keywords etc.

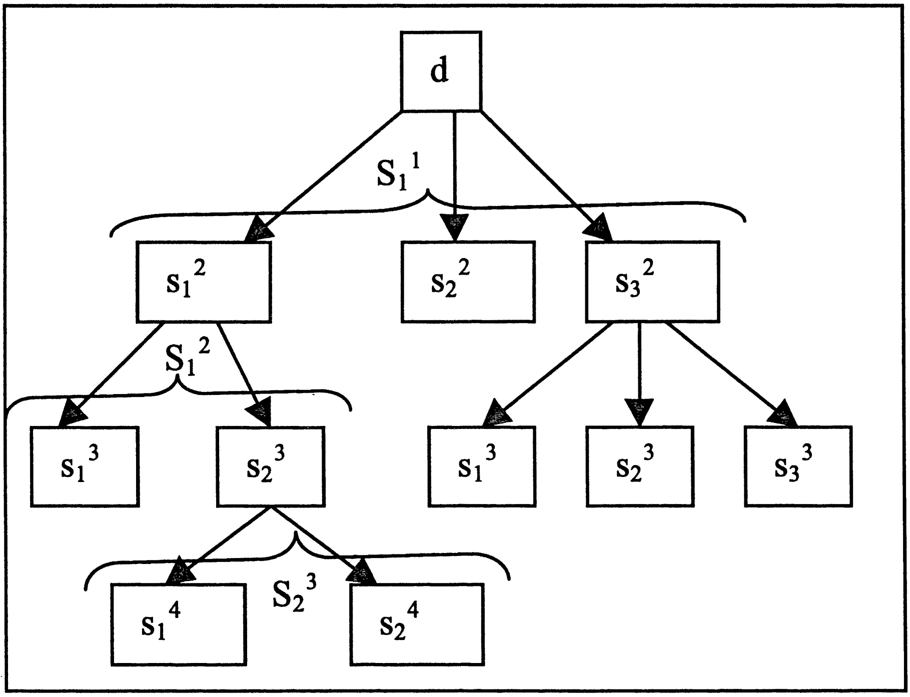

Fig. 1. A graph representation of 3-level structured document.

Some studies have faced the problem of defining data models of structured and semi-structured documents [5]. Some approaches extend the traditional inverted file data structure so as to keep track of the granularity of XML documents [8]. Others claim that the traditional inverted file data structure is not suited to this purpose and propose more sophisticated data models based on the object oriented paradigm [7, 18]. The common objective of these works is to try to increase the efficiency of the IRSs dealing with structured documents.

The objective of our paper is different, our intent is to define a flexible model for representing structured documents in an attempt to improve the effectiveness of the IRSs adopting it. Thus, we do not investigate which data model is more suited to store efficiently the representation of structured documents. We assume a graph-based representation of a collection of structured documents, such as that shown in Figure 1. A document can be characterized by a given number of sections, subsections and paragraphs, nested one into another. The leaves of the tree correspond to the paragraphs; paragraphs can be either pieces of unstructured text of variable length, or can be entities such as images or tables. The intermediate nodes correspond to structured sections having a variable number of child nodes.

Let us denote by s^i_k the kth section at level ith of a document k, and by S^i_k its structure, i.e. the set $S^i_k = \{s^{i+1}_1, s^{i+1}_2, \ldots, s^{i+1}_n\}$ of its n subsections, (child nodes) with $n = |S^i_k|$, i.e., the sections nested in s^i_k. In the following we will define a fuzzy representation of structured documents, by first considering documents with only one-level structure, then by generalizing this representation to the case of K-level documents.

3.1. A fuzzy representation of one-level structured documents

In many IR applications the documents in the considered archive are naturally structured in logical sections, such as *title, author* or *introduction*, when considering scientific papers [5, 9, 13, 14, 15].

In these documents, named one-level structured documents (see Figure 2), the occurrences of a given term have a distinct informative role depending on the subpart in which it appears. A single occurrence of the term in the title indicates that the paper is concerned with the concept expressed by the term, while a single occurrence in the *reference* suggests that the paper refers to other publications dealing with that concept. The role of each term occurrence depends then on the semantics of the subpart where it is located. This means that for defining an indexing function for structured documents the occurrences of a term contribute distinctly to the significance of the term in the whole document. Moreover, the documents subparts may have a different importance determined by the users' needs. For example, when looking for papers written by a certain author, the most important subpart is the *author name*, while when looking for papers on a certain topic, the *title, abstract,* and *introduction* subparts are preferred.

In [3] we have proposed an indexing model where the degree of significance of the index terms are computed by taking into account their occurrences in the different documents' sections to a distinct extent depending on both the semantics of the section and the user indications described in the following. At the level of query formulation the user can express her/his interpretation of the text. First, the archive is generated so that the system can recognize and manage the sections in which one wants to structure the documents. The sections are defined depending on the semantics of the documents. Then, during a retrieval phase, the user can formulate requests that specify conditions on the document structure. For example, the user can specify the distinct importance (preferences) of the sections and decide that a term must be present in all the sections of the document or in *at least a certain number* of them in order to consider the document relevant to her/his needs. The query evaluation mechanism performs a partial matching between the query and the fuzzy representation of one-level structured documents hereafter formally introduced.

A one-level structured document d has a structure S_1^1 consisting of n sections or paragraphs identified by $s_i^2 \in S_1^1 = \{s_1^2, \ldots, s_n^2\}$ where $i \in 1, \ldots, n$, and $n = |S_1^1|$.

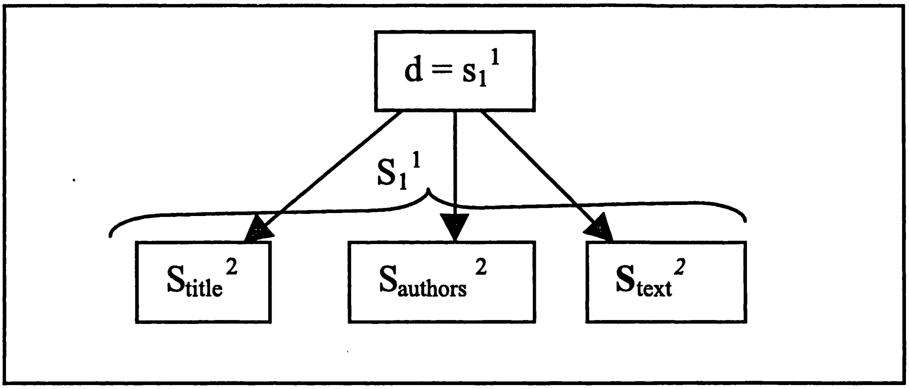

Fig. 2. A one-level structured document.

Formally, we represent a one-level structured document as a fuzzy binary relation R_d. For those not familiar with the fuzzy set notation, a fuzzy relation defined on the cartesian product of two sets A and B, $A \times B$, is denoted by $\sum_{(a,b)\in A\times B} \mu(a,b)/(a,b)$, where with each pair $(a,b) \in A \times B$ a membership value $\mu(a,b) \in [0,1]$ is associated to express the intensity of the relationship between a and b. At this point we can define R_d:

$$R_d = \sum\nolimits_{(t,s^2)\in T\times S_1^1} \mu(t,s^2)/(t,s^2). \tag{1}$$

The value $\mu(t,s^2) = F_{s^2}(d,t)$ expresses the significance of term t in section s^2 of document d, s^2 belongs to $S_1^1 = \{s_1^2, s_2^2, \ldots, s_n^2\}$ and T is the set of all the index terms. A function $F_{s^2} : D \times T \rightarrow [0,1]$ is then defined for each paragraph s^2. The definition of F_{s^2} is based on the semantics of section s^2 and can be specified by an expert during the indexing phase of the documents. For example, for textual sections containing short texts or formatted texts, such as *author* and *keywords*, a single occurrence of a term makes it fully significant in that section: in this case, we can for example assume that $F_{s^2}(d,t) = 1$, if t is present in s^2, $F_{s^2}(d,t) = 0$ otherwise. On the other side, for sections containing textual descriptions of variable length such as the *abstract* and *title* sections, $F_{s^2}(d,t)$ can be computed as a function of the normalized term frequency in the section like for example:

$$F_{s^2}(d,t) = tf_{ds^2t} * IDF_t$$

in which IDF_t is the inverse document frequency of term t defined as $IDF_t = \log(|D|/f_{td})$ where $|D|$ is the total number of documents in the collection D, and f_{td} is the frequency of term t in document d [24].
tf_{ds^2t} is defined as:

$$tf_{ds^2t} = \frac{OCC_{ds^2t}}{MAXOCC_{s^2d}}$$

in which OCC_{ds^2t} is the number of occurrences of term t in section s of document d and $MAXOCC_{s^2d}$ is a normalization parameter depending on the section's length so as not to underestimate the significance of short sections with respect to long ones [23]. This value can be for example computed as the frequency of the term with the highest number of occurrences in the section. In [3], in order to simplify the indexing procedure, this value is heuristically approximated: during the archive generation phase, the expert indicates the estimated percentage of the average length of each section with respect to the average length of documents ($PERL_{s^2}$). Given the number of occurrences of the most frequent term in each document d, $MAXOCC_d$, an approximation of the number of occurrences of the most frequent term in section s of document d is derived as:

$$MAXOCC_{s^2d} = PERL_{s^2} * MAXOCC_d.$$

3.2. A fuzzy representation of hierarchically structured documents

In this section we generalize the fuzzy representation defined in the previous section to the case of documents with a K-level structure. Let us first define the fuzzy representation of the jth section s^i_j at the ith level of a document d, denoted by $R_{s^i_j}$. We have to distinguish the case in which s^i_j identifies a leaf child node, for example a paragraph, from the case in which it is an intermediate node in the hierarchy that has a collection of child nodes associated with it. In the first case the paragraph s^i_j is represented as the fuzzy subset of the set T of index terms:

$$R_{s^i_j} = \sum\nolimits_{t \in T} F_{s^i_j}(d,t)/t \qquad (2)$$

in which t varies in T, and $F_{s^i_j}(d,t)$ is the membership value of t in s^i_j of document d, i.e. the significance degree, computed based on a function $F_{s^i_j}$ defined specifically for the paragraph s^i_j in one of the ways presented in the previous section.

When a document is completely unstructured, i.e. it is a single paragraph, it is represented as a fuzzy set of its index terms by formula (2) in which $s^i_j \equiv d$. When s^i_j is an intermediate node of the hierarchy, i.e., the structured section s^i_j, its representation is formally defined as a fuzzy binary relation:

$$R_{s^i_j} = \sum\nolimits_{(t,s^{i+1}) \in T \times S^i_j} \mu(t, s^{i+1})/(t, s^{i+1}) \tag{3}$$

in which $t \in T$, $s^{i+1} \in S^i_j = \{s^{i+1}_1, s^{i+1}_2, \ldots, s^{i+1}_n\}$ the set of the $n = |S^i_j|$ child nodes of node s^i_j and $\mu(t, s^{i+1})$ is the membership value of term t in section s^{i+1} of document d.

When the document d is a one-level structure, then $i = 1$ and $j = 1$, and formula (3) reduces to formula (1) defined in the previous section. The membership value $\mu(t, s^{i+1})$ expressing the significance degree of term t in section s^{i+1} is computed based on criteria specific for the section. If s^{i+1} is a paragraph, then $\mu(t, s^{i+1}) = F_{s^{i+1}}(d, t)$ expresses the significance of term t in section s^{i+1} of document d as defined in the previous section. If section s^{i+1} is structured, $\mu(t, s^{i+1})$ is computed based on an aggregation function $Ag : [0, 1]^n \to [0, 1]$ that combines the n significance degrees of term t in the immediate child nodes (subsections) of s^{i+1}.

$$\mu(t, s^{i+1}) = Ag\left(\mu(t, s^{i+2}_1), \mu(t, s^{i+2}_2), \ldots \mu(t, s^{i+2}_n)\right) \quad \text{in which } n = |S^{i+1}| \tag{4}$$

in which the values $\mu(t, s^{i+2}_j)$ are in their turn obtained as a result of an aggregation step of the significances of their child nodes of level $i+3$, unless they are leaves nodes with a significance degree which is the index term weight of term t in the considered section.

An aggregation step is then needed at each intermediate (non leaf) node s^i_j of the hierarchy because a single membership value $\mu(t, s^i_j)$, having the semantics of the significance degree of t in s^i_j must be computed. The value $\mu(t, s^i_j)$ is then used (with the other significance degrees of level i) to compute the significance of t in s^{i-1} at the higher hierarchy level $i - 1$.

The definition of the aggregation function Ag can be made based on several considerations. In the most general case,, a different Ag can be defined for each intermediate node, i. e. for each section s^{i+1}. This choice makes the indexing process more flexible. Another extreme solution is to assume a unique definition for all the structured sections: this makes the indexing process the more rigid, since sections having different semantics and nature are forced to adopt the same criterion for computing the significance degrees of their indexes. An intermediate solution is to define a different Ag function for each level of the hierarchy, from $i + 1$ to K. The choice should be made by considering the degree of homogeneity of the structured sections: for example, if they are all consisting of textual paragraphs it may be sufficient to adopt a unique definition of Ag. If they have a very heterogeneous structure so that some of them are multimedia sections containing images, captions, tables, and the other ones are textual paragraphs, two types of Ag function could be specified.

On the basis of formula (3) the fuzzy binary relation that represents a K-level structured document d is the following:

$$R_d = \sum\nolimits_{(t,s^2) \in T \times S^1_1} \mu(t, s^2)/(t, s^2)$$

where $\mu(t, s^2)$ is computed based on formula (4) by aggregating the significance degrees $\mu(t, s_1^3)$, $\mu(t, s_2^3), \ldots, \mu(t, s_n^3)$ of the direct subsections (or child nodes), and so on, recursively:

$$\mu(t, s^2) = Ag\left(\mu(t, s_1^3), \mu(t, s_2^3), \ldots, \mu(t, s_n^3)\right) \; n = |S^{i+1}| \tag{5}$$

and Ag is the aggregation function associated with node s^2.

During the indexing process, only the significance degrees of the index terms in the paragraphs of documents are computed and stored in the data structure, i. e., the values $F_s(d, t)$, in which s corresponds to a leaf node. The significance degrees $\mu(t, s)$ of structured sections (intermediate nodes) are computed at retrieval time. This allows one to implement a flexible indexing mechanism, since the aggregation functions Ag associated with the intermediate nodes and used to compute $\mu(t, s)$ can be specified dynamically.

In particular, as it will be described in the next section, given the n values $\mu(t, s_1^2)$, $\mu(t, s_2^2), \ldots, \mu(t, s_n^2)$ as arguments, the aggregation function Ag_d associated with the root node is used to compute the Retrieval Status Value of a document d with respect to a query term t. Ag_d can be specified by the user through a linguistic quantifier such as *most*, or *at least n*, so as to tune the retrieval ranking according to users needs. Further the user can express preferences on the documents' sections: the idea is that the significance degrees of terms in the most preferred sections must have a greater influence in determining the result of the aggregation function Ag. The query evaluation mechanism that allows a flexible selection of documents having a logical structure such as the one described above is presented below.

4. DEFINITION OF THE FLEXIBLE QUERY LANGUAGE FOR HETEROGENEOUS STRUCTURED DOCUMENTS

The fuzzy structured document representation presented in Section 3 provides a detailed description of both the documents' structure and the role of each index term in documents' sections. The basic information carried by this weighted representation can be employed to define an adaptive indexing procedure, in which the user can guide the term weight computation. This is done by means of the definition of a flexible query language, that is presented in this section, and which allows users to express some requirements concerning their "interpretation" of the document structure in their queries.

We assume that the collection is composed of heterogeneous documents, each one having its own specific structure. We assume that there are groups of homogeneously structured documents in a collection (journal articles, scientific papers, recipes, etc.). We note that the structure of each group of documents can be represented by a tree in which there are mandatory sections and optional sections. Mandatory sections are those present in all the documents of the group; optional sections are those present in a subset of documents in the group.

To simplify the query language and ease the user-system interaction, the tree structure of the documents is visualized through a user-interface. The mandatory sections are distinguished from the optional ones, which are marked by dotted edges (see Figure 3.)

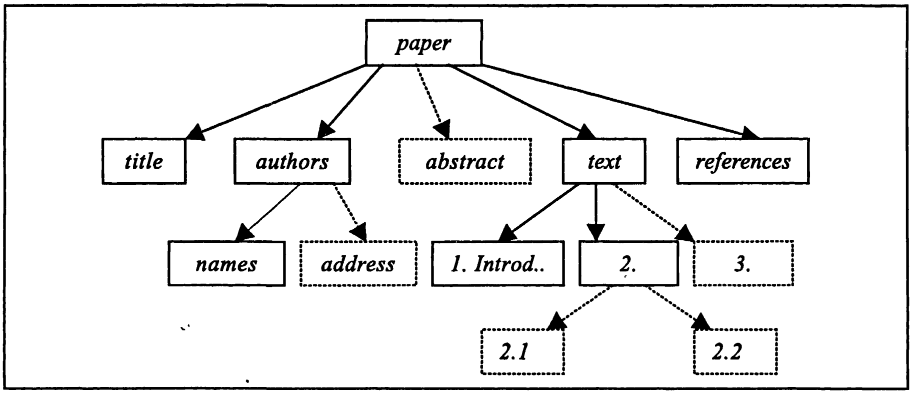

Fig. 3. Structure of a document in a collection of scientific papers.

In their query session users can express their preferences on the document structure in two steps. First they can specify the sections to be preferred in the document evaluation by directly clicking on the tree structure. The expression of the users' preferences on sections indicates that terms in the preferred sections should be considered more significant in determining the relevance of the documents to the query. The user can rank the sections in decreasing order of their preference (importance) to her/his information needs or an equal importance can be associated with the preferred sections.

The second phase consists of the formulation of a query that quantifies the number of sections in which a term should appear for the documents to be relevant. This means that if the presence of an index term is very important in order to consider the document relevant, the user can ask the term to appear in *at least one* section of a document. If on the other hand, the user desires to be more selective, he/she should ask for the presence of the index term in *most* or in *all* the sections.

As we will see below, the information provided by the user in this two-step interaction is combined by the query evaluation mechanism, with the aim of estimating the relevance of a document. In other words, the retrieval function evaluates the satisfaction of the two-level constraints in a query session by each document in the archive; the value of this function on a given document is the Retrieval Status Value (RSV) of that document and expresses the degree of relevance estimated by the system on the basis of the users indications.

The user can specify her/his preferences over the sections in two possible ways:

— by ranking the documents' sections in decreasing order of their perceived importance; it is assumed that section s_i is more important than section s_j iff $i < j$ (being i and j the positions of s_i and s_j respectively in the ordered list).

The sections that do not explicitly appear in this ranked list are considered meaningless to the user. This means that their contribution to the computation of the documents' relevance must be disregarded. The user can either specify different degrees of importance for the ranked sections or not. In this second case the numeric importance degrees $\alpha \in (0,1]$ are computed based solely on the sections ranking by applying the following formula: $\alpha_i = (k - i + 1)/k$ in which k is the cardinality of the ranked list and α_i is the importance degree computed for section si in the ranked list. With this definition of α_i the most important sections have an importance weight close to 1. The sections which are not listed are assigned an importance weight of zero.

— The user can choose a constant importance for the marked sections; in this case the marked sections are associated with a maximum importance weight of 1, while those that are not marked are assumed to have a zero weight.

Once the preferences over the sections have been expressed, a query is formulated in which the selection conditions on the structured documents' content are expressed at the level of the atomic query components, i. e. the search terms. We propose the definition of a query language as a generalization of the Boolean query language. The generalization consists in extending the atomic selection conditions, which in the Boolean query language are single terms.

We propose the definition of the following atomic component of the query (basic selection criterion):

$$t \text{ in } Q \text{ sections}$$

in which t is a search term, and Q is a linguistic quantifier such as *all*, *most*, or *at least k* (with k between 1 and the total number of the sections at the first level of the hierarchy). We limit the quantification to the first level of the hierarchy since in most the application domains, the sections that are semantically meaningful are those at the first level of the structure. The linguistic quantifier is used to associate with the root node an aggregation operator Ag that computes the Retrieval Status Value of document d to the query.

For further details on the definition of linguistic quantifiers see [28, 31]. In our context, linguistic quantifiers are associated with aggregation operators; in Section 4.1 we introduce the notion of Ordered Weighted Averaging Operators which can be adopted for using linguistic quantifiers as aggregation operators. In [28] a procedure for defining a weighted ordered averaging operator associated with linguistic quantifiers is defined.

4.1. Ordered Weighted Averaging Operators

In this section we will introduce the definition of the Ordered Weighted Averaging (*OWA*) operators that can be used to define aggregation functions specified by linguistic quantifiers such as *most of* or *at least n* [28]. We remind that in the computation of the RSV of a structured document thq aggregation function Ag_d associated with the root node is specified through a monotone increasing linguistic quantifier and is defined by an *OWA* operator.

An *OWA* operator of dimension n is an aggregation function $OWA : [0,1]^n \rightarrow [0,1]$ with a weighting vector $W = [w_1, w_2, \ldots, w_n]$ such that:

$$\sum_{j=1}^{n} w_j = 1, \quad \text{and} \quad w_j \in [0,1].$$

Further,

$$OWA(x_1, x_2, \ldots, x_n) = \sum_{j=1}^{n} w_j \operatorname{Max}_j(x_1, x_2, \ldots, x_n) \tag{6}$$

in which $\operatorname{Max}_j(x_1, x_2, \ldots, x_n)$ equals the jth biggest element of all the x_i [28]. For example, $\operatorname{Max}_1(x_1 = 0.8, x_2 = 0.5, x_3 = 1) = x_3 = 1$; $\operatorname{Max}_2(x_1 = 0.8, x_2 = 0.5, x_3 = 1) = x_1 = 0.8$; $\operatorname{Max}_3(x_1 = 0.8, x_2 = 0.5, x_3 = 1) = x_3 = 0.5$.

OWA operators are mean operators, that produce values which lie between those produced by the AND aggregation operator (min) and the OR aggregation operator (max). The degree of *orness* of an *OWA* aggregation operator expresses its closeness to the OR behavior, and it is defined as:

$$orness(W) = \left(\frac{1}{n-1}\right) \sum_{j=1}^{n} ((n-j) * w_j). \tag{7}$$

The *OWA* operator with the weighting vector $w^* = [1, 0, \ldots, 0]$ corresponds to the OR operator, i.e., the max. In this case, $orness(W^*) = 1$. The *OWA* operator with the weighting vector $W_* = [0, \ldots, 0, 1]$ corresponds to the AND operator, i.e., the min. In this case, $orness(W_*) = 0$.

An *OWA* operator can be defined with a weighting vector W modeling a linguistic quantifier such as for example *most of*, or *at least k* [29]. This definition of linguistic quantifiers allows interpreting them as aggregation operators. The elements of the weighting vector W of an *OWA* operator that represents a linguistic quantifier can be computed automatically as the number N of the instances to be aggregated varies. The linguistic quantifiers *all* and *at least one* correspond to the *OWA* operators with weighting vector W_* and W^* respectively. *OWA* operators with a soft behavior intermediate between the two extremes *all* and *at least one* can be defined as follows. First, by following Zadeh [30] the relative linguistic quantifier Q is defined as a fuzzy subsets with membership function $Q : [0,1] \rightarrow [0,1]$. The membership function representing a relative monotone non-decreasing quantifier Q is a monotone non-decreasing function, i.e., $Q(0) = 0$, $Q(1) = 1$, and $Q(x) \leq Q(y)$ for $x < y$. $Q(x)$ expresses the satisfaction in having $x\,\%$ of the elements satisfied.

Once Q has been defined, the *OWA* operator associated with it is determined by computing its weighting vector W: its N elements $w_i \in [0,1]$ are obtained as:

$$w_i = Q(i/N) - Q((i-1)/N) \quad \forall i = 1, \ldots, N. \tag{8}$$

In order to apply the *OWA* operator when distinct importance degrees $I_1, \ldots, I_N \in [0,1]$ are associated with its arguments, it is first necessary to modify the values $x_1, x_2, \ldots, x_n$ so as to increase the contrast between the most important arguments with respect to the least important ones. The modified degrees $a_1, \dots, a_N$ are obtained as follows:

$$a_i = [I_i \vee (1 - orness(W))] * (x_i)^{I_i \vee orness(W)} \tag{9}$$

in which W is the OWA weighting vector, and $\vee$ is defined as the max operator. Then, the OWA operator is applied to the modified values $a_1, \dots, a_N$.

4.2. Query evaluation

A legitimate query of the proposed flexible query language is a Boolean expression such as the following:

(t_1 in Q_1 sections) AND (t_2 in Q_2 sections) OR (t_3 in Q_3 sections).

The degree of satisfaction, RSV(d, t), of a document d with respect to a selection condition (t in Q sections) is obtained by combining the single significance degrees of the document's sections at the first level of the hierarchy through the aggregation function, the OWA operator, identified by the linguistic quantifier Q, possibly by taking into account their importance weights $\alpha_s \in [0, 1]$.

In the simplest case of one-level structured documents, the significance degrees to be aggregated by the OWA operator are already stored in the data structure since they have been computed during the indexing phase.

If the user has not specified any preference on the documents' sections, the sections are assumed equally important, and the OWA_Q operator is directly applied to the significance degrees:

$$\mathrm{RSV}(d, t) = OWA_Q \left(F_{s_1^2}(d, t), F_{s_2^2}(d, t), \dots, F_{s_n^2}(d, t) \right).$$

If the sections have a distinct importance, the significance degrees are first modified in order to increase the "contrast" between the contribution due to more important sections with respect to less important ones by applying formula (9). Then the OWA operator is applied to the modified values.

In the general case of K-level structured documents, the evaluation of the query is based on a recursive procedure based on a bottom up traversal of the tree representing the document structure. It starts from the intermediate nodes whose child nodes correspond to the tree leaves, and goes up the tree hierarchy until the root node is reached. At each intermediate node s_j^i its significance degree is computed through the aggregation function Ag associated with the node s_j^i. The Ag function is applied to the values $\mu(t, s_1^{i+1}), \mu(t, s_2^{i+1}), \dots, \mu(t, s_n^{i+1})$ with $n = |S_j^i|$ so as to emphasize their contributions proportionally to their importance weights $\alpha_1^{i+1}, \alpha_2^{i+1}, \dots, \alpha_n^{i+1}$. The values $\mu(t, s_1^{i+1})$, $\mu(t, s_2^{i+1}), \dots, \mu(t, s_n^{i+1})$ are the significance degrees associated with the child nodes of s_j^i and have been computed at the preceding step. When the root node is reached, the significance degrees of the sections at the first level are aggregated, possibly modified by their importance weights $\alpha_s \in [0, 1]$ computed on the basis of the ranking of users preferences as has been illustrated in Section 4.1 and indicated as:

$$\mathrm{RSV}(d,t) = OWA_Q\left(\mu(t,s_1^2), \mu(t,s_2^2), \dots, \mu(t,s_n^2)\right).$$

5. AN EXAMPLE

In the following we briefly sketch an example of query evaluation specifying a selection condition on the structure of scientific papers. We have considered a collection of one-level structured documents with six sections: *title, author's names and affiliation, abstract, keywords, text* and *references*. The following query is evaluated:

$$q = \boldsymbol{t} \text{ in } \textbf{most} \text{ sections}$$

with the following ranking of the sections: *title, keywords, abstract, text, references, authors*.

Table 1 shows the significance degree of the term $\boldsymbol{t}$ in each section where it occurs. These degrees are obtained using the indexing process; since the *title, keywords*, and *authors* sections are short texts, μ_{title} and $\mu_{keywords}$ are defined so as to take values in $\{0,1\}$. After estimating that the text section takes up on average 70% of the documents' length, and the *reference* section is around 10%, μ_{text} and $\mu_{reference}$ are defined as described in Section 3.

$$most(x) = \begin{cases} 0 & \text{for } \quad x \le 0.5 \\ \frac{10}{3}x - \frac{5}{3} & \text{for } \quad 0.5 < x < 0.8 \\ 1 & \text{for } \quad x \ge 0.8 \end{cases}$$

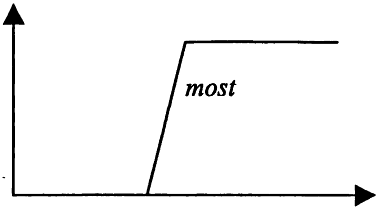

Fig. 4. Representation of the linguistic quantifier *most*.

Table 1. Significance degrees of the term $\boldsymbol{t}$ in the sections of four documents.

$\mu_{\cdot}(\cdot, \boldsymbol{t})$	*title*	*authors*	*keywords*	*abstract*	*text*	*reference*
d_1	1	0	1	0.8	0.4	0.2
d_2	0	1	0	0	0	0.8
d_3	0	0	1	0	0.8	0.6
d_4	1	0	1	0	0.6	0.8

The linguistic quantifier most is represented in Figure 4 and its definition based on the notion of fuzzy set of the unit interval [31] is the following one:

The weighting vector of the OWA_{most} operator is obtained by applying formula (8) with the number of the sections $N = 6$, i. e.:

$$w_i = most(i/6) - most((i-1)/6) \quad \forall i = 1, \dots, 6$$
$$\text{then} \quad W_{most} = [0, 0, 0, 5/9, 4/9, 0].$$

The degree of *orness* of the OWA_{most} operator is obtained by applying formula (7) with $N = 6$, i. e.:

$$orness(W) = \left(\frac{1}{n-1}\right) \sum_{j=1}^{n} ((n-j) * w_j) = 14/45 = 0.3111.$$

This low value of the *orness* reflects the fact that the semantics of *most* is closer to that of *all* (AND) than to that of *at least one* (OR).

The importance scores of the sections α_i, $i = 1 \dots 6$ are obtained from the sections' ranking as follows:

$$\alpha_i = \frac{6 - i + 1}{6}$$

in which i is the position of the section and 6 is the number of the sections within the ranked list. We then obtain the following importance scores:

$$\alpha_{title} = 1; \quad \alpha_{authors} = 1/6; \quad \alpha_{abstract} = 2/3;$$

$$\alpha_{keywords} = 5/6; \quad \alpha_{text} = 1/2; \quad \alpha_{references} = 1/3.$$

The modified values of the significance degrees of the term t in each section of the documents in Table 1 are reported in Table 2. They are obtained by applying formula (9) defined in Section 4.1. The last column reports the retrieval status value of the documents with respect to the query.

Table 2. Modified significance degrees of the term ***t***.

μ	*title*	*authors*	*keywords*	*abstract*	*text*	*reference*	RSV(d, t)
d_1	1	0	0.83	0.59	0.43	0.4	**0.4**
d_2	0	0.68	0	0	0	0.61	**0**
d_3	0	0	0.83	0	0.61	0.58	**0**
d_4	1	0	0.83	0	0.53	0.63	**0.16**

By this simple example it can be seen that documents d_1 and d_4 are retrieved with d_1 having a higher relevance than d_4. This reflects the fact that ***t*** is significant in the most important sections of document d_1.

6. CONCLUSIONS

In this paper a fuzzy indexing model of structured documents has been proposed, together with a query language that allows users to tune the representation of documents based on their search perspectives. The definition of adaptive indexing mechanisms constitutes a step towards the design of flexible IRSs, dealing with documents having heterogeneous structures. In particular, the proposed model allows users to specify their preferences on the documents' sections at retrieval time, so the significance of the index terms in documents is determined in a "dynamic" way by taking into account the users view of the documents.

(Received June 13, 2000.)

REFERENCES

[1] A. Bookstein: Fuzzy requests: an approach to weighted Boolean searches. J. Amer. Soc. Inform. Science *31* (1980), 240–247.

[2] G. Bordogna and G. Pasi: A fuzzy linguistic approach generalizing Boolean IR: a model and its evaluation. J. Amer. Soc. Inform. Science *44* (1993), 2, 70–82.

[3] G. Bordogna and G. Pasi: Controlling retrieval through a user adaptive representation of documents. Internat. J. Approx. Reason. *12* (1995), 317–339.

[4] G. Bordogna and G. Pasi: Linguistic aggregation operators of selection criteria in fuzzy information retrieval. Internat. J. Intelligent Systems *10* (1995), 233–248.

[5] Y. Chiaramella and A. Kheirbek: An integrated model for hypermedia and information retrieval. In: Information Retrieval and Hypertext (M. Agosti and A. Smeaton, eds.), 1996, pp. 136–176.

[6] D. A. Buell D. H. and Kraft: Threshold values and Boolean retrieval systems. Inform. Process. Management *17* (1981), 127–136.

[7] V. Christophides et al: From structured documents to novel query facilities. In: Proc. ACM SIGMOD Internat. Conf. on Management of Data. ACM Press, Minneapolis 1994.

[8] D. Florescu, I. Manolescu and D. Kossmann: Storing and querying XML data using an RDBMS. IEEE Data Engineering Bulletin *22* (1999), 3, 27–34.

[9] H. Kim and S. Cho: Structured storage and retrieval of SGML documents using GROVE. Inform. Process. Management *36* (2000), 643–657.

[10] R. Krovetz and W. B. Croft: Lexical ambiguity and information retrieval. ACM Trans. Information System *10* (1992), 2, 115–141.

[11] G. J. Klir and T. A. Folger: Fuzzy Sets, Uncertainty and Information. Prentice Hall PTR Englewood Cliffs, 1998.

[12] D.H. Kraft, G. Bordogna and G. Pasi: An extended fuzzy linguistic approach to generalize Boolean information retrieval. J. Inform. Sciences Appl. *2* (1995), 3, 119–134.

[13] M. Lalmas and I. Ruthven: Representing and retrieving structured documents using the Dempster–Shafer theory of evidence: Modelling and Evaluation. J. Documentation *54* (1998), 5, 529–565.

[14] I. Macleod: Storage and retrieval of structured documents. Inform. Process. Management *26* (1990), 2, 197–208.

[15] A. Molinari and G. Pasi: A fuzzy representation of HTML documents for information retrieval systems: In: Proc. IEEE Internat. Conf. on Fuzzy Systems, New Orleans 1996.

[16] C. V. Negoita: On the notion of relevance in information retrieval. Kybernetes *2* (1973), 3, 161–165.

[17] C. D. Paice: Soft evaluation of Boolean search queries in information retrieval systems. Information Technology: Research Development Applications *3* (1984), 1, 33–41.

[18] Y. Papakonstantinou, J. Widom and H. G. Molina: Object exchange and heterogeneous information sources. In: Proc. IEEE Internat. Conf. on Engineering, Birmingham 1996.

[19] F. Paradis and C. Berrut: Experiments with theme extraction in explanatory texts. In: Proc. II Internat. Conf. on Conceptions of Library and Information (CoLIB 2), Copenhagen 1996, pp. 13–16, 433–446.

[20] J. Perez–Carballo and T. Strzalkowski: Natural language information retrieval: Progress report. Inform. Process. Management *36* (2000), 155–178.

[21] A. Rao et al: Query Processing in TREC-6. Inform. Process. Management *36* (2000), 179–186.

[22] N. Sager: Natural Language Information Processing. Addison Wesley, 1981.

[23] G. Salton, E. Fox and H. Wu: Extended Boolean information retrieval. Comm. ACM *26* (1983), 12, 1022–1036.

[24] G. Salton and M. J. McGill: Introduction to modern information retrieval. McGraw–Hill Internat. Book Co., 1984.

[25] K. A. Sparck Jones: Automatic Keyword Classification for Information Retrieval. Butterworths, London 1971.

[26] K. A. Sparck Jones: A statistical interpretation of term specificity and its application in retrieval. J. Documentation *28* (1972), 1, 11–20.

[27] C. J. van Rijsbergen: Information Retrieval. Butterworths, London 1979.

[28] R. R. Yager: On ordered weighted averaging aggregation operators in multi criteria decision making. IEEE Trans. Systems Man Cybernet. *18* (1988), 1, 183–190.

[29] R. R Yager and J. Kacprzyk (eds.): The Ordered Weighted Averaging Operators: Theory and Applications. Kluwer, Dordrecht 1997.

[30] L. A. Zadeh: Fuzzy sets. Inform. and Control *8* (1965), 338–353.

[31] L. A. Zadeh: A computational approach to fuzzy quantifiers in natural languages. Comput. Math. Appl. *9* (1983), 149–184.

Dr. Gloria Bordogna and Dr. Gabriella Pasi, Istituto per le Tecnologie Informatiche Multimediali CNR, via Ampère 56, 20131 Milano. Italy.
e-mails: gloria.bordogna@itim.mi.cnr.it, gabriella.pasi@itim.mi.cnr.it