ELECTRONIC **WORKSHOPS IN COMPUTING**

Series edited by Professor C.J. van Rijsbergen

Ian Ruthven (Ed)

Miro'95

Proceedings of the Final Workshop on Multimedia Information Retrieval (Miro '95)
Glasgow, Scotland
18-20 September 1995

Paper:

Using Abductive Inference and Dynamic Indexing to Retrieve Multimedia SGML Documents

Adrian Müller and Said Kutschekmanesch

Published in collaboration with the
British Computer Society

BCS

Springer

Using Abductive Inference and Dynamic Indexing to Retrieve Multimedia SGML Documents

Adrian Müller and Said Kutschekmanesch
GMD (German National Research Center for Information Technology)
IPSI (Integrated Publication and Information Systems Institute)
Dolivostr. 15, D-64293 Darmstadt, FRG
email: {amueller,kutschek}@darmstadt.gmd.de

Abstract

The retrieval of complex multimedia items such as SGML-structured texts can be facilitated by means of a formal representation of knowledge about these data. These information sources must be aggregated dynamically at the time of query processing. In this paper, an interactive, probabilistic retrieval system is proposed, comprising an extended Bayesian network, a multimedia indexing component and an abductive retrieval engine. The inference process exploits and controls the index structure of the network. The prototype has been tested on a collection of SGML structured dictionary articles. An example is presented in the last section of the paper.

1 The need for structured information retrieval

Indexing, parsing and other information retrieval techniques can be integrated to provide high quality browsing and retrieval systems. We consider formalisms for the identification of different document types and components as a platform for a comprehensive approach to structured document retrieval. This analysis must be guided by the syntactic structure and the associated semantic meaning. The key idea is to employ the syntactic structures of SGML data the way human authors do. This papers describes an information retrieval experiment to access a collection of SGML structured biographies in the domain of art and artists by means of an abductive inference and a dynamic indexing module.

Applying IR techniques to structured documents can be done by using the document structure to guide the retrieval process, as we have pointed out recently [MT94]. First experiments on the evaluation of different retrieval methods [Wil94] have shown that knowledge of the structure of documents can improve retrieval effectiveness. With a semi-automatic method of mapping from document structures to control techniques for an appropriate indexing and retrieval system, one can take full advantage of the information structures without the need to model all aspects of the domain knowledge.

In the following we give an overview on SGML and corresponding IR techniques, and we investigate the utilization of probabilistic information retrieval as a means to capture the syntactic and vague semantic knowledge which can be derived from a document description. First, we introduce an extended inference network to cope with multiple indexes, which are computed from the text structure and the domain model. As Poole (c.f.[Poo93]) has shown, probabilistic Horn abduction and Bayesian networks are semantically equivalent under certain restrictions. Hence, in the following section we discuss the abductive retrieval engine of the system MIRACLE (MultImedia concept Retrieval bAsed on logiCaL query Expansion), which grounds on a representation shared with the indexing network. Thus, the inference process of MIRACLE can combine conceptual and syntactical information to map high-level query statements to appropriate positions in the probabilistic index network. The subsequent section introduces the automatic indexing system MAGIC (Multimedia-based Automatic Generation of Indexes and Clusters), which extracts syntactic elements from structured multimedia documents with taking into account semantic knowledge of the domain. Finally, we show an illustrative trace of a query on an SGML dictionary of art and artists.

2 SGML and IR

The Standardized Generalized Markup Language *SGML* [ISO86] is intended as a means for the free interchange of information between people as well as between computer environments. There has been significant growth in the number of available tools such as editors, parsers and database support mechanisms, and hence the number and size of available SGML-structured documents is increasing as well.

An *SGML Document-Type-Definition (DTD)* is essentially a grammar specifying the logical structure of documents of a certain type. Examples of document types are dictionaries, journals, and articles. Each document type defines a set of valid markup tags, which can be included within the text. These tags identify the individual logical document components, the so-called *elements*. Chapters, footnotes, embedded frames (e.g., an excurse on a certain person), locations, picture links and the like are examples of elements.

Human authors annotate these components with semantic meanings (e.g., typed attributes and links like 'supports'), which are defined very imprecisely by means of comments, or which are grouped together as a kind of meta-definition (e.g., HyTime). However, there exists neither a unified semantic standard nor a common retrieval semantic. There are several initiatives to promote the uniform use of SGML and related markup systems.

The Text Encoding and Interchange group (TEI) recently has produced the 1,300 page TEI Guidelines for Electronic Text Encoding and Interchange, providing guidelines for uniform SGML encoding. The Document Style Semantics and Specification Language (DSSSL), the companion to SGML for formatting and transformation, has been largely re-written. But what is still missing is a way to map an arbitrary information need to a heterogeneous collection of data.

A number of query languages for structured documents, and particularly SGML documents, have been developed [CACS94, BCK+94]. One of the key problems is whether the exact document structure must be known in order to formulate queries. This can, to a certain degree, be accomplished by extending traditional query languages (like SQL) with a path-operator allowing abstraction from individual element types and individual structures.

The identification of logically coherent parts of a document is another approach to increase retrieval effectiveness. Several techniques, derived from linguistic and text-pragmatic research, have been successfully integrated. On the one hand, applying these techniques requires large-scale meta knowledge, either in terms of sophisticated parsing techniques or in terms of an additional domain model to identify the proper textual components. On the other hand, statistical techniques for passage retrieval have proven to be useful [Cal94], but only for uniformly structured document collections.

None of the approaches discussed are applicable to SGML documents in the general case. Instances (i.e. documents) of a uniform DTD grammar differ in size and structure of components, thus disabling unguided statistical techniques. In contrast to this, there is a variety of domains with different semantics for which similar SGML structures can be found, since SGML is intended for the encoding of different types (but not different meanings) of texts. Unfortunately, an adequate modeling of the domains would be too expensive to perform using current knowledge representation techniques.

There are two reasons why the retrieval of documents via concepts is important. On the one hand, if the database contains multimedia documents, a global index of high-level concepts is needed, because a user often searches documents about a special topic and not an item of the document which is maybe a graphical explanation.

On the other hand, one has also to keep in mind that a structured document is the result of a creative process, in which the author (or editor) wants to express an intention. In a structured document, this intention can be found in the formal description of the document's content. We will return to this aspect in the following sections.

3 Indexing and Combining Information

In the next sections, we will sketch our implemented prototype. The retrieval effectivness of an IR system can be improved if a user is informed about the structure of the documents(c.f.[Wil94]). We show the way in which the MIRACLE system decomposes and aggregates structural and semantic information to provide more and detailed feedback to a user.

Figure 1 shows an overview of the system architecture. We will introduce the two main modules (the retrieval engine of MIRACLE and the indexing system MAGIC) and explain the contents of the shared knowledge bases, shown in between.

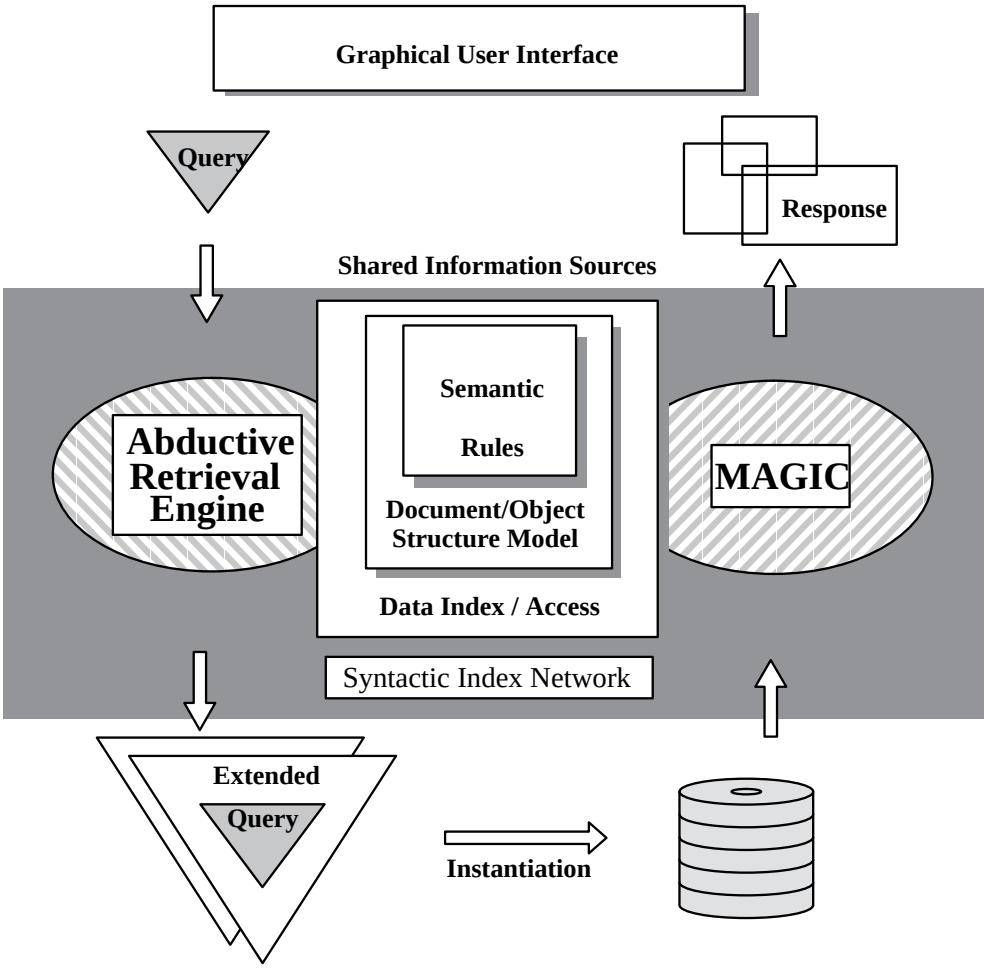


Figure 1: System architecture

3.1 Indexing and Probabilistic IR

There are two different approaches in the field of probabilistic information retrieval. The inference network approach (e.g. [Pea88]), which forms a node from each term and each document, is able to retrieve freetext queries by propagating them through the network and producing an approximated probabilistic relevance value for each document. The rule-based approach (e.g. [FHL+91], [Tze94]) tries to identify important terms from documents and infers the factual knowledge-base from the analysis. Most of the existing probabilistic information retrieval systems work by indexing one medium (mostly textual information), or processing each medium independently. In contrast, we believe that various media as well as both the textual and non-textual parts of a multimedia document should be treated in common. To handle this problem, we designed an automatic indexing system that combines the network and the rule-based approach in one system. Our solution to multimedia indexing combines extraction of domain and case-specific knowledge and fulltext indexing: We represent the system's knowledge by means of an inference network and we extract additional information using a domain-specific rulebase and a rule interpreter which is capable of interpreting the rulebase while indexing.

Our rule interpreter is able to gather the SGML-tagged information by means of the associated rulebase, we interpret the rules describing the SGML-tagged information as indexing functions. Non-textual media can be connected to the index network by combining additional indexing functions which cover these media (e.g. statistical analysis for voice patterns, signatures for pictures, etc.). A rule contains a condition and a result. A condition is formed by an expression

which contains predicates and logical operators. These predicates are capable of addressing selectable areas of the document. This is done by looking for the tags which are described in the DTD. A result is formed by a special predicate $indexTerm()$ which describes the semantic interpretation of the condition. The instances of the resulting predicate $indexTerm()$ and the rule identifier which fired the instance are added as an access key for query processing to the network.

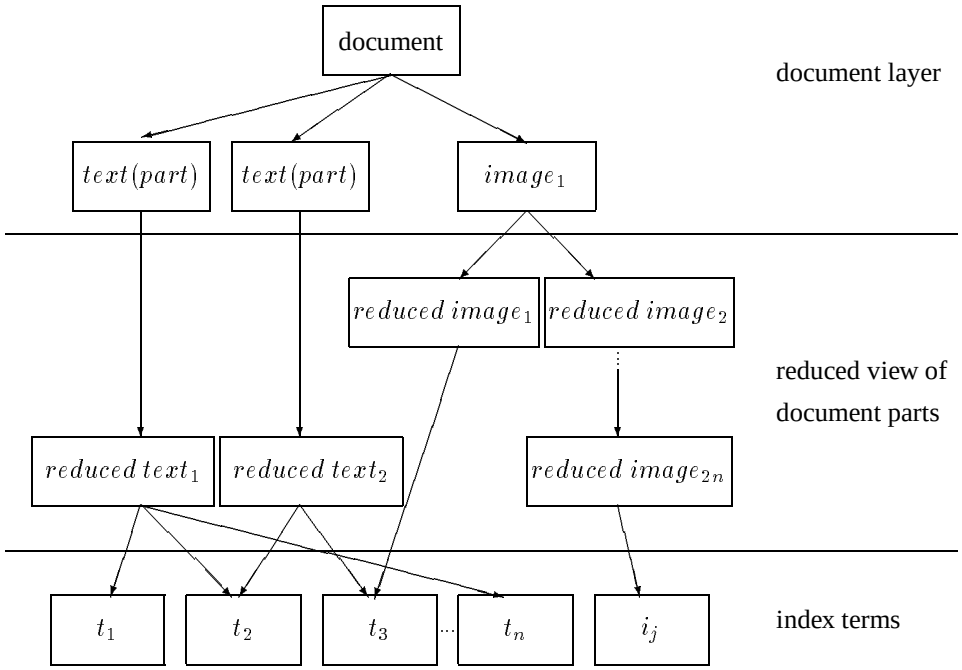


Figure 2: Structure of the index network

We use a Bayesian network, c.f. [TC90], [TC91], which we expanded to contain three layers (see figure 2). It is used as the first information source (see figure 1) while processing a query. The hierarchical levels of the network may be interpreted as increasing levels of abstraction.

Using statistical information gathered from the individual data (documents, pictures, etc.) we compute approximate initial probabilistic weights for the nodes (e.g. term frequency (tf) and inverse document frequency (idf) [Sal86]). The first (document) layer contains the multimedia documents as a summary of their monomedial parts. A document may contain a set of textual nodes thereby representing text passages. The second (reduced) layer contains a reduced view of each document part. A document part can result in a set of reduced nodes. The third (term) layer contains the index terms. The index terms include the terms which are contained in the document and the terms which were instances of the predicate $indexTerm()$.

When a query is to be processed, MIRACLE is able to activate and modify the weights of the rules which were evaluated beforehand. In this way, the system transforms index terms into concepts representing the context the user is interested in. The query terms are matched against the third layer. Starting from all active nodes, the algorithm proceeds bottom-up through the hierarchy until a document is reached, thereby combining several information sources into a relevance assignment.

3.2 MIRACLE - the retrieval engine

Complex data such as multimedia items, medical files, SGML or HTML structured texts have a rich syntactic structure and associated operators (e.g. players, viewers) but they typically lack an explicitly defined semantics. One way to capture the content and intended semantics of these data is to compute the access parameters and methods by means of an inference process which computes a mapping of all available information (query operators, data types and structures, user model) to a directly executable expansion of the query.

A widely-used technique for reasoning in information retrieval (c.f. e.g. [Nie92, Hes92, MSST93]) is *deductive* inference, mostly within first-order or probabilistic logic. Such systems assign a truth value to a given query by computing the deductive closure of a given theory (a set of axioms, stored in a database, and a set of rules) and checking whether the query is an element of this closure. On a semantic level of abstraction, this is similar to Datalog-based information systems, as well. However, in contrast to the interpretation of Datalog programs, general inference mechanisms allow a richer expressiveness of rules and hence they do not delimit their model of truth to be the extensional closure of facts. A notational variant of standard semantics of first-order calculi are descriptive (or terminologic) logics (DL) in the tradition of KL-ONE [BS85]. IR approaches using DL (c.f. e.g. [MSST93]) primarily use subsumption hierarchies to provide a conceptual model of the domain and they classify a query expression with in that hierarchy to match it against data items.

Deduction-based approaches like the ones sketched above - which define their domain model in a top-down manner - need to provide an accurate modeling of their domain and have to face the fact that changes in any part of the theory might lead to inconsistencies: ad-hoc changes in higher levels of the rule base might sometimes even lead to parts of the data being irretrievable.

In the context of logical IR, the general goal of an inference sequence is to map from high-level query statements to appropriate retrieval operations on index terms. *Abductive reasoning*, the inference process we use in the retrieval engine of the MIRACLE system, builds upon a bottom-up inference process, i.e., one needs to model the basic properties of the domain and the rules for the aggregations of plain data items. Then, the task of abduction can be roughly described as 'process the formula $doc \rightarrow query$ [vR89] from right to left', i.e., an abductive inference process will try *directly* to find a way from a given query to the available data.

Abduction generates a set of explanations which imply the consequence (the query). These formulae may be regarded as system-generated interpretations of the user's information need in terms of database (or domain) structure and contents. As a consequence of applying abductive reasoning, this process yields not only a query expansion on the level of search terms, as e.g. in thesaurus-based systems, but also produces different possible readings of the query which may differ in their meaning on both the semantic and structural level.

To put it another way: Abductive reasoning tries to find almost every solution for a given problem (here: how to find information in a database), but it does not require a global and consistent modeling of the content of the database. This property of abduction is illustrated by its most prominent application area: fault analysis in complex systems. This consistency constraint is the crucial precondition of pure deductive retrieval systems (c.f. [Seb94]) and it is worthwile to overcome it: The required consistency of knowledge bases and terminological logics will not get scaled-up for real world sized applications.

An inference calculus needs to be defined in several aspects. First we introduce the basic inference step. Second, the formal theory is defined and rewritten for the application IR. The final step is to specify the constraints for the inference mechanism, i.e., to design the retrieval engine.

The basic inference step of abduction can roughly be described as a kind of symmetric inversion of Modus Ponens.

$$\frac{\varphi \rightarrow \omega, \omega}{\varphi}$$

Read: Given $\varphi \rightarrow \omega$ and ω is observed, then φ is a reason for ω

A short example, which contrasts the classical example of deductive inference, might illustrate the basic abductive inference step in a toy world: All humans are mortal ($\varphi \rightarrow \omega$). Socrates is dead ($\omega$). Hence, Socrates is (or was) a human being (φ).

The relations of abduction within an information retrieval system are implicational and not necessarily causal. In general, an abductive calculus will find several possible explanations with respect to a given set of data and a query formulation. Since the union of all explanations does **not** need to be consistent, one might refer to each explanation φ_i as a feasible *hypothesis.*

Definition of an abductive logic: Given a theory $\mathcal{T}$ and a sentence ω, which needs to be explained in terms of $\mathcal{T}$, abduction will yield a set of hypotheses φ so that

$$\mathcal{T} \cup \varphi \vdash \omega \qquad holds.$$

Abductive logic offers a straightforward way of processing and mediating concepts over a given domain, assuming it has been indexed properly and the theory $\mathcal{T}$ reflects the inherent properties of the data and the information system which holds the data. A request $Query$ is a description of a concept the user is looking for. Let $\omega = Query$ be an

existentially quantified sentence combining elements of $\mathcal{T}$. Then we can rewrite the abductive reasoning process for information retrieval systems as:

$$\mathcal{T} \cup Concept \vdash Query,$$

Read: Find all concepts from which the query can be derived. The notion of $\vdash$ is a semantic relation, which must hold for all instances that can be derived from $\mathcal{T} \cup Concept$.

Abductive systems need to know when to stop searching for further explanations. The basic principle is to define a set $\mathcal{A} \in \mathcal{T}$ to be the set of *abducible* sentences. All hypotheses must ground on elements of $\mathcal{A}$. By setting $\mathcal{A}$ to be the content of the accessible database(s), we guarantee the usefulness of the inference process. To be more precise:

Terminating the inference process: *The set $\mathcal{A}$ of abducible sentences is the collection of atomic axioms. Each axiom corresponds directly to an information item via a computational access method within the database of the system.* In this case, $\mathcal{A}$ is defined to be the set of accessible atomic information items in MAGIC, i.e., the union of all concepts in the index network.

The elements of $\mathcal{A}$ might get combined in virtually all possible permutations, because the inference process will try to find all feasible solutions (for each given query). Since this set of results will be very large for a real-world database, the construction of $\mathcal{A}$ cannot be done a priori (at the time of indexing of the data collection), but instead must be computed dynamically at query time.

One should note that each potential combination of elements of $\mathcal{A}$ needs to be executable at query time. The pseudo concept $indexTerm()$ is a syntactic trigger to identify the elements (i.e. the index terms) which are covered by a rule. Hence, the connections from the inference engine to the probabilistic indexing sub-system need to be parameterized by terms, rules or combinations thereof at the time of query execution.

A static indexing function supposes only one way (or a limited number of ways) to combine relevant information, i.e., query terms or other activated basic items. Since optimal solutions might require a large variety of access modes, the structure of static indexing modules (which force an application to pre-select the indexing technique) restricts the inference engine to a limited number of solutions. Slightly parameterizable indexing functions (like the passage retrieval mode of the INQUERY system) add a certain degree of freedom to the retrieval system (here: to adjust the average passage length dynamically with respect to the document type) and hence increase the *quality* of the results [Cal94]. In the next section, we will demonstrate our design of a *dynamic indexing function*, which is based on the already introduced three-layered Bayesian network.

Summing up, the theory $\mathcal{T}$ has to reflect the properties of the subsystem(s) which are connected to the retrieval engine via $\mathcal{A}$. The theory $\mathcal{T}$ is organized as a hierarchy of three layers (semantic rules of the domain model, representation of database structure, and $\mathcal{A}$, mapping from structures to access methods - c.f. figure 1), where each layer is grounded on the top axioms of the underlying level and the lowest level is implemented as a set of executable relations.

Definition of $\mathcal{T}$: *A logical theory $\mathcal{T}$ is defined over the language $\mathcal{L}$ of well-formed first-order logic formulae, built from variables, constants and predicates.* A *rule* is of the form: $p_1 \wedge \ldots \wedge p_i \rightarrow a_j \wedge \ldots a_n$ where each a_i is a predicate with an arity greater than zero. Preconditions p_i can either be refined in subsequent rules (i.e. becoming a normal predicate a) or they can be omitted to indicate basic concepts (e.g. see rule (1) below). The constants of $\mathcal{T}$ are defined by the set of index terms. Variables and predicates range over subsets of index terms.

Now the process of abductive retrieval is defined as:

1. A query is given as an intensional description of an information need. It is an existential quantified sentence, constructed from elements of $\mathcal{T}$.

2. The query is reformulated with respect to $\mathcal{T}$, so that the inference process ends up with a set of non-contradicting hypotheses which map the user-defined concepts to the basic data.

3. Each hypothesis is valid in the sense that its extension (the content of the data) will satisfy the user's query with respect to its interpretation in $\mathcal{A}$. The notion of satisfaction is based on the semantics of the logical theory applied.

Note that this is a procedural *model of truth, as opposed to* declarative *models, which are used frequently in deductive systems.*

This approach possesses some desirable properties. Usually, a query can be interpretated in more than one way. Since abduction finds *sets* of possible solutions (sets of instantiated hypotheses), it shows an intrinsic advantage as a retrieval technique: the inherent ambiguity of queries is reflected in a straightforward manner in the retrieval process

by offering mutual distinct hypotheses to the user, whereas multiple solutions for *one* hypothesis H_i are captured by evaluating all models $\|H_i\|$ by means of the executable relations in $\mathcal{A}$.

In the next section, we will introduce the multimedia indexing component MAGIC, which is the major component of $\mathcal{A}$. It is designed to provide a high flexibility for the combination of information. Thereby the complete retrieval system can fully exploit the query processing techniques of MIRACLE.

3.3 MAGIC

Most electronic documents today contain standard publication notations (such as SGML or HTML). This presentational information can often be used to identify the title, passages, the name of the author or important words (like locations or dates). The automatic indexing system MAGIC is designed to model semantic concepts from the available syntactic information. If the document collection uses a DTD, an indexer can define a rulebase which will be used for this document collection.

We want to point out that we do not model the world or an entire domain. We only model the DTD to gain as much useful information from the document structure as possible. In addition to this we add rules which combine results from different media. The rulebase can be modified by an experienced user. Therefore, it is possible to use different rulebases for different databases which may also have different document structures.

To index a selected document, we compute a reduced set of data. In case of a text, we delete stopwords and stem the remaining terms. The processed abstraction of a document part is stored as a node, which represents the reduced view in the network. Depending on the type of media, different numbers of reduced nodes per document part may be possible in the second layer (e.g. an image may be indexed with different feature extraction algorithms like shape or color histogramm). In the last step, we add the resulting terms as nodes to the third layer of the network. After the net has been synthesized, the rule interpreter evaluates the correlation between the term and document nodes of the network. A correlation is described by the SGML structure and a term, which express the meaning of the entity (e.g. the semantic entity of the profession 'painter' is expressed by the term 'painter'). Different entities can be combined with operators to describe more complex correlations (e.g. an $artist$ has the $profession$ 'painter' and is still 'alive').

After building the network, the indexing system is ready to process a request. A request consists of a set of activated rules, their rule weights, and a set of query terms. The query terms are matched with the inverted terms of the third layer. From these entry points the system proceeds upwards and computes the weight of every document containing one or more of the query terms.

We use the following equations to compute the belief value of a document d_f:

In the following formulas, b_d denotes the default belief of a node. It is used when no precomputed belief can be found via the active rules. Normally, the default belief is set to 0.4, as was proposed by Turtle/Croft [TC90]. d_f represents the document that is currently indexed. t_j represents the query term in the jth position.

- compute weight

$$tr(j, d_f) = \sum_{\substack{r_l \in AR, \\ t_j \in eval(r_l)}} weight(r_l) \quad \text{with } t_j \in query$$

  $tr(j, d_f)$ denotes the weight of the query term t_j based on all activated rules that have a connection to the (selected) document d_f. AR denotes the set of activated rules. $eval(r_l)$ represents the outcome, if rule l is evaluated. r_l denotes the (activated) rule l, and $weight(r_i)$ is the given rule weight.

  This formula expresses that if a query term is indexed by more than one active rule within one document, the rule weights are treated equally.

- normalize rough belief

$$b_n(t_j, d_f) = \begin{cases} \frac{tr(j,d_f)}{\max_{t_k \in query} tr(k,d_f) + b_d}; & \text{for } tr(j, d_f) \neq 0 \\ b_d; & \text{for } tr(j, d_f) = 0 \end{cases}$$

  $b_n(t_j, d_f)$ denotes the normalized belief of the term t_j when document d_f is activated before the query terms are processed. We interpret $b_n(t_j, d_f)$ as a precompiled belief of the inference network.

  This formula guarantees that the belief is a value in the open interval between zero and one.

- compute document belief

$$P(d_f \mid t_j) \approx b_n(t_j, d_f) + (1 - b_n(t_j, d_f)) * ntf_{ij} * nidf_i$$

ntf_{ij} denotes the normalized term frequency ([TC90]); $nidf_i$ is the normalized inverse document frequency [TC90]

This formula approximates the probability that a document d_f is relevant to the query term t_j.

This is done for all query terms. From the term weights, MAGIC approximates the belief that a document is relevant for the whole query.

$$P(d_f \mid t_j t_{j+1} ... t_k) \approx \frac{P(d_f \mid t_j) + ... + P(d_f \mid t_k)}{k - j}$$

This formula combines the results from single-term queries to an approximated probability of a larger query.

Using these formulas, the system is able to modify the ranking of the results. The system selects a set of concepts (according to activated rules) and varies their weights with respect to the belief default or the weight of other active rules. The weights are selected using heuristical analysis. This step can be interpreted as the system trying to 'train' the network.

4 Processing a query

In the following sections, we will describe the prototype, which currently covers a subset of a dictionary of art and artists. As we have mentioned in previous sections, one can take advantage of the given syntactic and semantic knowledge about structured documents if that information can be related to the information need of the user.

The documents of the collection contain markup information which distinguish between special terms (e.g., person names, towns, countries, professions, dates) and ordinary terms. There are several document types (biographies, surveys etc.), which differ in their DTD, and there are various semantic links between entities of the domain (e.g., 'person A created picture B'). We will not discuss the structure of the data in detail but refer to their description which was presented at EP'94 [RMF94].

The collection has been indexed with the MAGIC system. MAGIC and the abductive retrieval engine of MIRACLE share a knowledge base (see figure 1) about indexing methods, data access procedures and syntactic and semantic knowledge of the collection, i.e., they mutually share a theory $\mathcal{T}$. This theory builds the cognitive model for users of the system.

At query time, a query statement (a formula in $\mathcal{T}$) is interpreted on the intensional level. The system provides feedback (the different abductive hypotheses) which shows how it will process the specified (or requested) query arguments. When the user has selected the proper interpretation(s), MIRACLE tries to instantiate the corresponding formulas, thereby executing the procedural axioms of $\mathcal{A}$. Now the user inspects the extensional truth of the hypotheses. Thus she can distinguish between query misconceptions and a lack of data; or in other cases she can identify underspecified query parameters.

The set of indexing rules is the starting point for the development of the theory $\mathcal{T}$. Each index term is mapped to at least one basic concept of the theory. Thus, we can ensure that a query is interpreted by using all possibilities of the MAGIC system. For a previous prototype of our system, we developed a similiar rule base to cope with the expressiveness of the INQUERY [CCH92] system. Since this system has been designed to process all index terms efficiently, the number of potential conceptualizations of terms is much smaller as was consequently the size of the required rule base.

On top of these domain-biased rules we added semantic rules like consistency constraints for concept-concept relations and semantic links between basic items. Another part of the theory contains an extra reasoning module, which can compute set-of relations (e.g., relations like *town_in_country()*), numerical comparisons like $1920 < 1937$ and the like). We give examples of the theory in the following sections and show the use of the additional reasoning module.

4.1 Shared Information Sources

In the following we give a few examples taken from our theory T to show the principles according to which a domain like a dictionary of art can be modeled and searched with the MIRACLE system. The basic predicates (like *artist()*, *profession()*, etc.) are designed in a one-to-one manner according to the grammar rules of the original SGML documents (e.g. the content of the tag field '$< artist >$' contains all potential instantiations of the predicate *artist()* - see rule(1)). During the indexing phase, the preconditions of a rule (e.g. $textType$) are checked, and for each qualifying document, chapter etc. the terms contained therein get assigned to zero, one or many appropriate concept(s). This $1 : n$ relation can be accessed by the syntactic trigger (pseudo-concept) $IndexTerm$. Hence, an element of the shared knowledge base is defined as a triplet:

$(rule, List\ of\ IndexTerm(s), weight)$

with $rule \in \mathcal{T}$, $IndexTerm$ can be a constant from, or a variable ranging over arguments of predicates from $\mathcal{T}$, and $weight$ is a heuristically assigned real number in $[0 \ldots 1]$.

The rules use the '$\rightarrow$' relationship to reflect the conditions which must hold for a valid document, e.g., the DTD requires that the '$< profession >$' tag must not be empty (see rule (4)). Additional knowledge was derived by inspecting the different document types (e.g. see rule (2)) or by adding a few straightforward reasoning modules to the system ('an artist starts working later than the time he/she was born' - see rule (3)). In the following, we will illustrate the rules which contribute to the example given in the next section.

1. $(\ \forall\ A: \quad \rightarrow artist(A),\ indexTerm(A),\ 0.8\ )$
   All terms A are marked as $artist$, if this term can be identified as (the name of) an artist. (The name of the artist is tagged in his/her biography). There is no precondition, because this rule can be evaluated independently at indexing time. While processing a query, the concept $artist$ can be abduced without further requirements.

2. $(\ \forall\ D \quad \exists\ A,\ T, \quad A \in T:\ doc(D, A, T) \wedge textType(biography, D) \rightarrow artist(A),\ null,\ null\ )$
   This rule expresses the fact that the *Dictionary of Art* is a collection of different document types. Each biography is concerned with exactly one artist (as opposed to a survey article) and it contains a collection of terms T. This rule does not deal with index terms.

3. $(\ \forall\ A,\ F \quad \exists\ B:\ artist(A) \wedge greater(B, F) \wedge birthdate(A, B) \rightarrow from\_year(F),\ null,\ null\ )$
   The birth date of an artist is a heuristic estimation of which time period this artist belongs to (to be more precise: the starting point of that period - see the use of greater() in the precondition part of this rule). This rule does not deal with index terms.

4. $(\ \forall\ A \quad \exists\ P:\ artist(A) \rightarrow profession(P),\ indexTerm(P),\ 0.6\ )$
   Each artist has (at least) one profession P. The relevance of the term P in the biography about the artist A will be marked, thus distinguishing the relevance of P from fortuitous occurrences in other documents. $profession$ denotes a set like $(painter, sculptor, architect, \ldots)$

Summing up, the shared knowledge base reflects static (indexing) and dynamic (querying) aspects of the domain under consideration, which are computed in the corresponding modules (index network, retrieval engine) of the system.

4.2 An Example Session

Now suppose an information need like: *"What do we know about the 20th century; in which countries do we find abstract art?"*. The user can express this somewhat underspecified query statement in MIRACLE by means of a WWW-based form widget, shown in figure 3. The query is translated into the internal representation. It is an existentially quantified formula of first-order logic:

$Q = \exists\ C:\ country(C), about(abstract\ art), profession(painter), from\_year(1900)$.

Please note that the user did not ask directly to search for an $artist$ or some other complex concept, e.g. a $document$. She leaves the task of relating the basic query parameters completely to the inference engine. Now the abductive mechanism explores all possibilities to combine the attributes in a consistent and executable way by building all proof graphs, chaining backwards from the actual query parameters. Our inference algorithm features a similar design to the suggestions of Pople, but it differs from [HP73] in its treatment of variables. Instead of standard skolemization we use an expanded term unification mechanism to maintain the scope of universally or existentially quantified variables, which might be bound during synthesis of partial proofs.

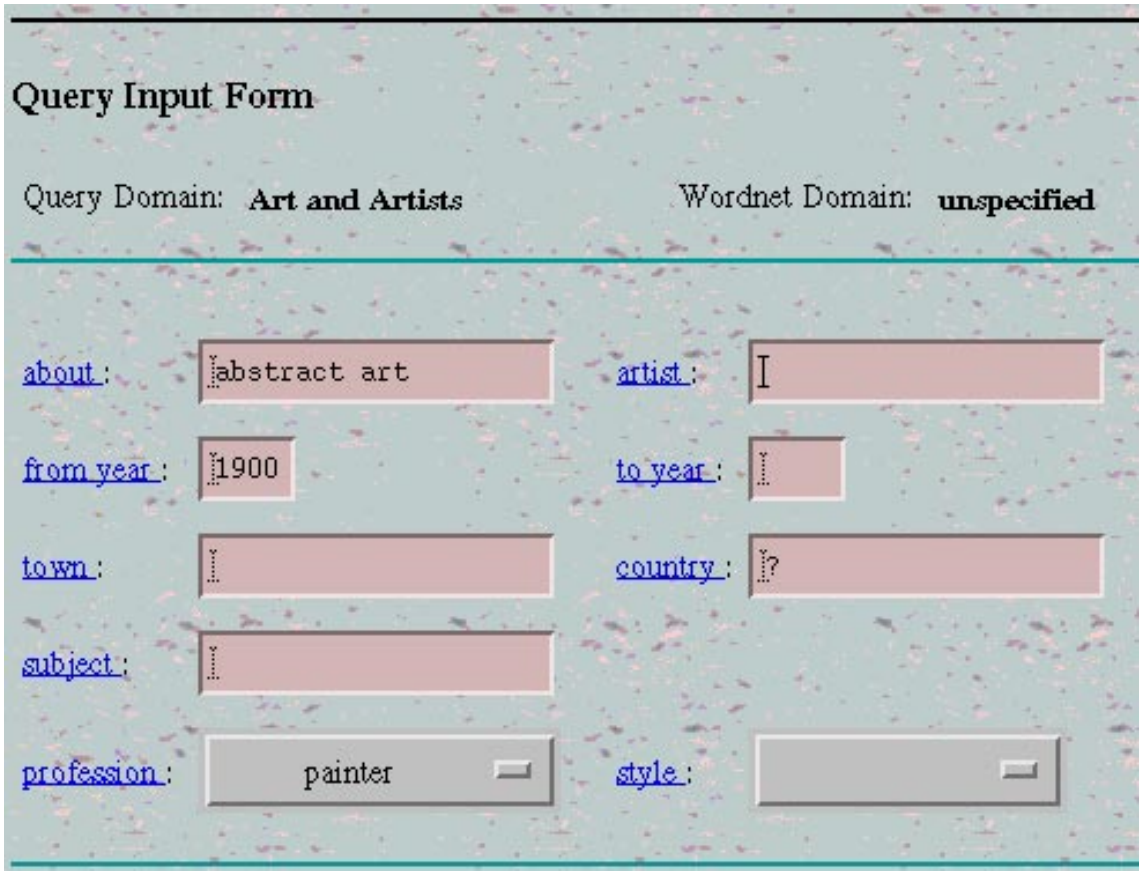


Figure 3: MIRACLE/WWW query form

The abductive procedure concludes that several isolated query attributes of the query Q can be grouped together by establishing joins from and to the central concept $artist()$. Thus, the abduced hypothesis for query interpretation Q_1 is:

$$H_1 = \exists\ A,\ B\ :\ artist(A), greater(B, 1900)$$

The rules used for an interpretation are considered to be *active* for this hypothesis. For example, since the user activated the concept $artist()$, MIRACLE restricts via rule (2) the document type to biographies. The query interpretation suggests further that, within this interpretation, $profession$ has to be related to the artist (see rule (4)) the document is about. Each active rule increases the weights for the corresponding set specified by $indexTerm()$. E.g., the term 'painter' achieves an increased belief value, if it denotes the profession of the artist under consideration.

Each hypothesis forms a directed and acyclic graph (DAG), which is presented in an interactive graphical interface. Figure 4 shows solution Q_1 of MIRACLE's query interpretation. Solid boxes indicate query parameters, dotted boxes either contain intermediate rule fragments or (like $greater(B, 1900)$) abduced predicates. The proof DAG is layouted with the Kamada algorithm [KK89]. This simulated annealing algorithm groups together related attributes, so that the intended semantic interpretation becomes more readable than straightforward (e.g. hierarchical) layouted inference proof structures. The arrows indicate the flow of information, i.e., they show how the aggregration of basic terms will be computed. Basic factual data like $country(C)$ are treated as requested (existential quantified) attributes. The results are derived by instantiating the variable 'C' (which stands for all known countries) with all strings which occur inside the corresponding SGML-text patterns (begin and end tags).

A second, alternative query interpretation Q_2 would suggest computing $country$ by looking for pictures or sculptures of artist A and the place they have been created, or their current locations. That means MIRACLE would

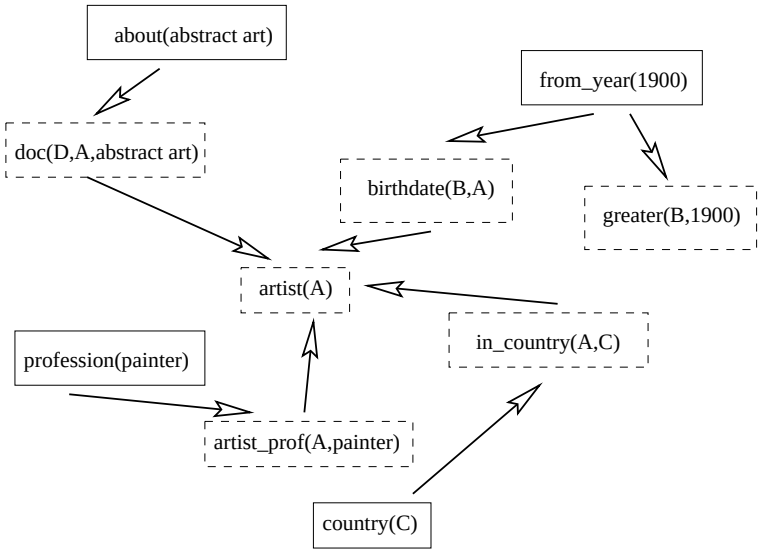


Figure 4: The intensional query interpretation Q_1

trace the curriculum vitae of an artist (by looking up all entries for all pieces of art, created by that artist), instead of recognizing his or her place of birth.

Now that all textual query terms ('abstract, art, painter') are interpreted (which means that the corresponding rules are active), the MAGIC component can assign the relevance values to the document collection. This is the first step in the calculation of $||Q_1||$, the model (i.e. the *extension*) of Q_1.

Intermediate computation of $||Q_1||$

Rel.Val.	**Doc.-ID.**	**Artist**
0.49560	A000101	Max Ackermann
0.48455	K031498	Noboru Kitawaki
0.47858	M028710	Armando Morales
0.47499	B001603	Francis Bacon
0.46957	D007552	Marcel Duchamp
0.46870	B002208	Peter Behrens
.....		

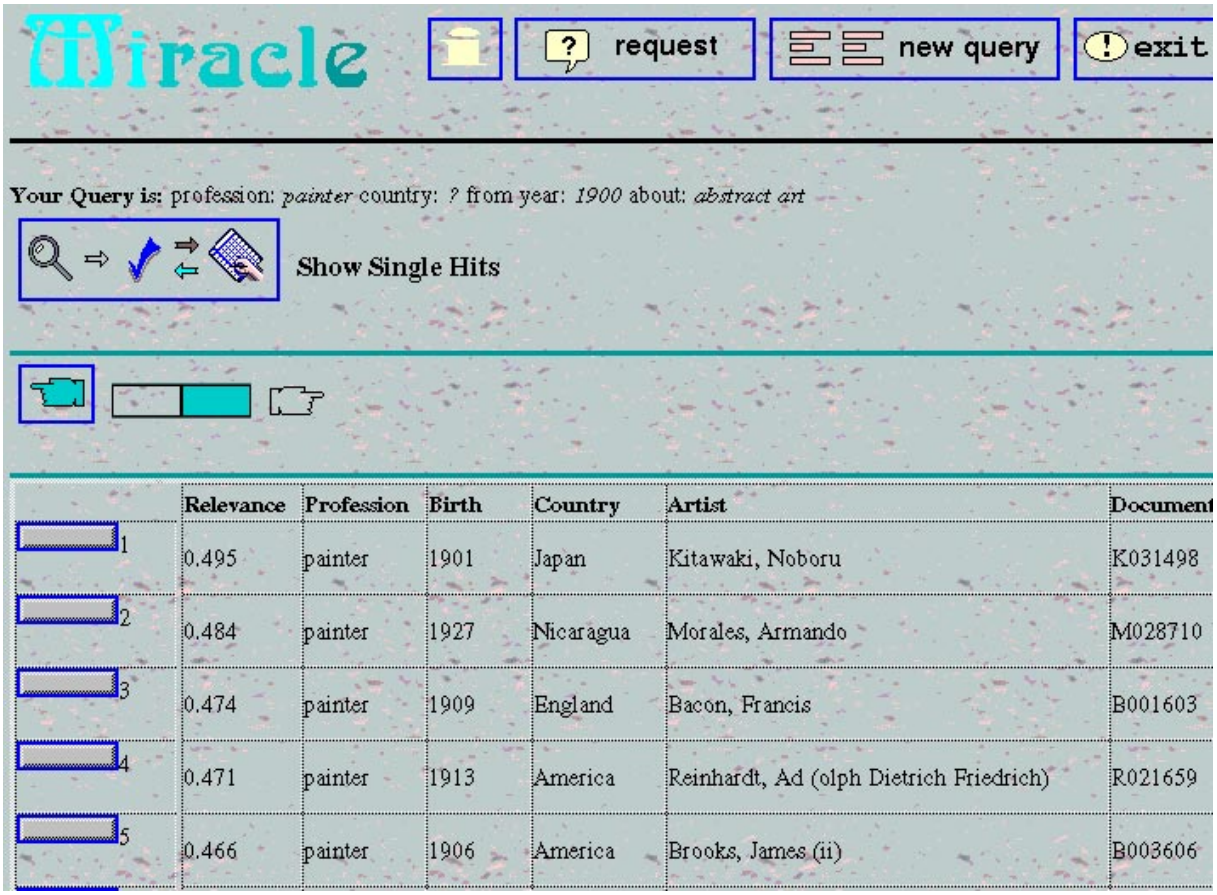
In the next step, MIRACLE instantiates the remaining procedural elements of Q_1. The time period $from\_year$ is mapped to an artist's birth date (see again figure 4). Since the $greater(B, 1900)$ relation does not hold for some of the intermediate solutions (Max Ackermann was born in 1877, Duchamp in 1887, Behrens in 1868) these solutions are rejected.

The remaining objects are assigned to the corresponding variables. Finally, the functionally dependent attributes ($country$) are calculated or checked by means of the corresponding relation (e.g. *in_country(Morales, Nicaragua)*, *artist_prof(Morales, painter)*. The instantiated formula for the top ranked result is:

$country(Japan) \wedge greater(1901, 1900) \wedge doc(K031498, Kitawaki, abstract\ art) \wedge$
$birthdate(1901, Kitawaki) \wedge artist(Kitawaki)$

This and the other instantiations of formula Q_1 define the extensional truth for the given query under this interpretation. It can be browsed item by item or summarized as a table, which is shown in figure 5. Please note that the result table has been synthesized on-the-fly from the active concepts of the current query interpretation, i.e., MIRACLE automatically assigned a column for each active concept and fills in the proper instantiations for each row.

A note on the **implementation**: MIRACLE/WWW has been implemented in C and SWI-PROLOG. The extended



	Relevance	Profession	Birth	Country	Artist	Document
1	0.495	painter	1901	Japan	Kitawaki, Noboru	K031498
2	0.484	painter	1927	Nicaragua	Morales, Armando	M028710
3	0.474	painter	1909	England	Bacon, Francis	B001603
4	0.471	painter	1913	America	Reinhardt, Ad (olph Dietrich Friedrich)	R021659
5	0.466	painter	1906	America	Brooks, James (ii)	B003606

Figure 5: MIRACLE/WWW result table

Bayesian network and MAGIC are written in Smalltalk. The rules which are used by MAGIC are located in a textfile. MAGIC loads the rules from the file using an interpreter integrated into MAGIC. The system runs on SYSV and BSD-UNIX machines. It currently contains approximately three thousand biographies in the domain of art. The SGML structure is captured by some 50 rules. The prototype, running on a SPARC-20, can be accessed at http://www-cui.darmstadt.gmd.de/~amueller/miracle.html.

5 Conclusion

Combining an abductive inference process, a dynamic indexing function and an appropriate multimedia index network facilitates the retrieval of complex and structured data. A domain model has been derived bottom-up from the given grammar (DTD) of a collection of SGML documents. It is formulated as a rule base, which does not need to cover all aspects of the real word for this domain, but mimics the features which the editors of the DTD have regarded to be important for the collection (e.g., artists have a certain profession, a place and time of birth, etc.). Thus, the modelling of functional dependencies, data types and the aggregation of information are consistent to the set of data we are talking about.

This rule base is shared between a query reformulation module (the retrieval inference engine of MIRACLE) and a multimedia indexing component (the MAGIC system). Thus, MAGIC can interpret the rules as indexing functions, which in turn can be adjusted at query time by the MIRACLE system. The prototype has been implemented and tested

for a collection of SGML-structured documents about art and artists.

Further research will try to achieve a weighted abductive retrieval system, i.e. to assign and compute subjective weights in the rule base. A promising method is to modify the rule-weights by the query context and by relevance feedback. In [TGMS96] we have shown how a dialogue manager can control the retrieval dialogue. This dialogue component will function as the missing link, specifying and negotiating the query context and how a user can enter relevance feedback in complex retrieval situations.

5.1 Acknowledgements

We are grateful to Bruce Croft for stimulating discussions on our ideas about abductive IR, and we would like to thank him and his group at UMASS for making the INQUERY system available. Special thanks go to Reginald Ferber, Barbara Lutes, Adelheit Stein and Ulrich Thiel for their comments and suggestions on the work described in this paper.

References

[BCK+94] G.E. Blake, M.P. Consens, P. Kilpeläinen, P.-A. Larson, T. Snider, and F.W. Tompa. Text / Relational Database Management Systems: Harmonizing SQL and SGML. In *Proceedings of the First International Conference on Applications of Databases*, Lecture Notes in Computer Science. Springer Verlag, June 1994.

[BS85] R. J. Brachmann and J. G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9:171–216, 1985.

[CACS94] V. Christophides, S. Abiteboul, S. Cluet, and M. Scholl. From Structured Documents to Novel Query Facilities. In *Proceedings ACM SIGMOD*, May 1994.

[Cal94] James P. Callan. Passage-level evidence in document retrieval. In *Proceedings of the 17th. Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 302–310. Springer, July 1994.

[CCH92] James P. Callan, W. Bruce Croft, and Stephen M. Harding. The INQUERY retrieval system. In *Proceedings of the Third International Conference on Database and Expert Systems Application*, pages 78–83, Valencia, Spain, 1992. Springer-Verlag.

[FHL+91] N. Fuhr, S. Hartmann, G. Lustig, M. Schwantner, K. Tzeras, and G. Knorz. AIR/X - a rule-based multistage indexing system for large subject fields. In *Proceedings of the RIAO '91*, pages 606–623, Barcelona, Spain, 1991.

[Hes92] M. Hess. An incrementally extensible document retrieval system based on linguistic and logical principles. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, June 1992.

[HP73] Jr. H.E. Pople. On the mechanization of abductive logic. In *Proceedings of the 3rd International Joint Conference on Artificial Intelligence*, pages 147–151, 1973.

[ISO86] Information Processing - Text and Office Systems - Standardized Generalized Markup Language (SGML), 1986. International Organization for Standardization.

[KK89] T. Kamada and S. Kawai. An algorithm for drawing undirected graphs. *Information Processing Letters*, 31:7–15, April 1989.

[MSST93] C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In *Proceedings of the 16th Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, pages 298–307, June 1993.

[MT94] Adrian Müller and Ulrich Thiel. Query Expansion in an Abductive Information Retrieval System. In *Proceedings of RIAO*, New York, N.Y., October 1994.

[Nie92] Jian-Yun Nie. Towards a probabilistic modal logic for semantic-based information retrieval. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, June 1992.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, 1988.

[Poo93] David Poole. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, 64:81–129, 1993.

[RMF94] L. Rostek, W. Möhr, and D. Fischer. Weaving a web: the structure and creation of an object network representing an electronic reference work. In V. Quint C. Hüser, W. Möhr, editor, *Proceedings of the Fifth International Conference on Electonic Publishing, Document Manipulation and Typography*, pages 495–505. John Wiley & Sons, Ltd., April 1994.

[Sal86] G. Salton. On the use of term associations in automatic information retrieval. In *Proceedings of Coling '86*, pages 380–386, 1986.

[Seb94] F. Sebastiani. A Probabilistic Terminological Logic for Modelling Information Retrieval. In *Proceedings of the 17th Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, pages 122–130, July 1994.

[TC90] H. Turtle and W.B. Croft. Inference networks for document retrieval. In *Proceedings of the 13th Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, September 1990.

[TC91] Howard R. Turtle and Bruce W. Croft. Efficient probabilistic inference for text retrieval. In *RIAO '91*. ACM, 1991.

[TGMS96] U. Thiel, Jon Atle Gulla, Adrian Müller, and Adelheit Stein. Dialogue strategies for multimedia retrieval: Intertwining abductive reasoning and dialogue planning. In *Miro edition of Workshops in Computing*. this volume, 1996.

[Tze94] Kostas Tzeras. *Wissensbasierte interaktive Indexierung*. PhD thesis, Technische Hochschule Darmstadt, 1994.

[vR89] C.J. van Rijsbergen. Towards an information logic. In *Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, June 1989.

[Wil94] Ross Wilkinson. Effective Retrieval of Structured Documents. In *Proceedings of the 17th. Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 311–317. Springer, July 1994.